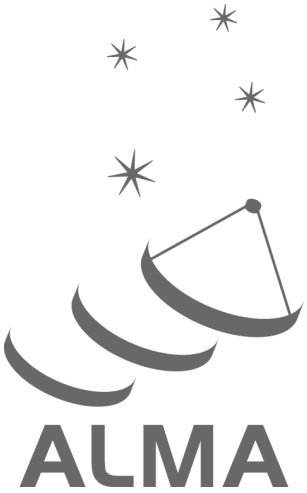
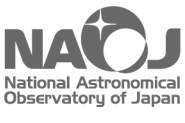
ALMA Science Pipeline Reference Manual

Pipeline CASA 4.3.1





www.almascience.org

ALMA, an international astronomy facility, is a partnership of Europe, North America and East Asia in cooperation with the Republic of Chile.

User Support:

For further information or to comment on this document, please contact your regional Helpdesk through the ALMA User Portal at **www.almascience.org**. Helpdesk tickets will be directed to the appropriate ALMA Regional Center at ESO, NAOJ or NRAO.

Revision History:

Version	Date	Editors
1.0	September 2014	Pipeline Team
2.0	August 2015	Pipeline Team

In publications, please refer to this document as:

ALMA Pipeline Team, 2015, ALMA Science Pipeline Reference Manual, Version 2.0, ALMA

Table of contents

1 Purpose and scope

The purpose of this document is to describe the tasks available for the calibration and imaging of interferometry datasets using the ALMA Science Pipeline.

2 Pipeline Task Types

There are 5 types of Pipeline tasks. This document provides descriptions of the task types: h_, hif_ and hifa_.

Table 1: Pipeline Task Types

Task pre-fix	Task type	Description
h_	Common tasks	Pipeline tasks used in the calibration and imaging of both interferometry and single-dish datasets
hif_	Interferometry common tasks	Pipeline tasks used in the calibration and imaging of both ALMA and EVLA interferometry datasets
hifa_	Interferometry ALMA tasks	Pipeline tasks used in the calibration and imaging of ALMA interferometry datasets only
hifv_	Interferometry EVLA tasks	Pipeline tasks used in the calibration and imaging of EVLA interferometry datasets only
hsd_	Single-dish tasks	Pipeline tasks used in the calibration and imaging of single-dish datasets only

3 Pipeline Context

The Pipeline state is stored in its context e.g. which calibration tables need to be used at each stage. Several Pipeline tasks are associated with initialising and editing (rarely needed) the context.

4 Pipeline Task List

The tasks used in the current ALMA standard recipes (casa_pipescript.py and casa_piperestorescript.py) are shown in bold. Tasks which do not form part of the standard recipes are currently experimental.

Table 2: Common Tasks

	Task name	Description
1	**h_init**	Initialise the interferometry pipeline
2	**h_resume**	Restore a save pipeline state from disk
3	**h_save**	Save the pipeline state to disk
4	**h_weblog**	Open the pipeline weblog in a browser

Table 3: Interferometry Common Tasks

	Task name	Description
1	hif_antpos	Derive an antenna position calibration table
2	**hif_applycal**	Apply the calibration(s) to the data
3	hif_atmflag	Flag channels with bad atmospheric transmission
4	hif_bandpass	Compute bandpass calibration solutions
5	hif_bpflagchans	Flag deviant channels in bandpass calibration
6	**hif_cleanlist**	Compute clean map
7	hif_clean	Compute clean map
8	**hif_export_calstate**	Save the pipeline calibration state to disk
9	**hif_exportdata**	Prepare interferometry data for export
10	hif_gaincal	Determine temporal gains from calibrator observations
11	hif_import_calstate	Import a calibration state from disk
12	hif_importdata	Imports data into the interferometry pipeline
13	**hif_lowgainflag**	Flag antennas with low or high gain

14	**hif_makecleanlist**	Compute list of clean images to be produced
15	hif_normflux	Average calibrator fluxes across measurement sets
16	**hif_rawflagchans**	Flag deviant channels in raw data
17	**hif_refant**	Select the best reference antennas
18	**hif_restoredata**	Restore flagged and calibration interferometry data from a pipeline run
19	**hif_setjy**	Fill the model column with calibrated visibilities
20	hif_show_calstate	Show the current pipeline calibration state

Table 4: Interferometry ALMA Tasks

	Task name	**Description**
1	**hifa_bandpass**	Compute bandpass calibration solutions
2	**hifa_bpsolint**	Compute optimal bandpass calibration solution intervals
3	**hifa_flagdata**	Do basic flagging of a list of measurement sets
4	**hifa_fluxcalflag**	Locate line regions in solar system flux calibrator spws
5	**hifa_fluxdb**	Connect to flux calibrator dataset
5	**hifa_gfluxscale**	Derive flux density scales from standard calibrators
6	**hifa_importdata**	Imports data into the interferometry pipeline
7	hifa_linpolcal	Compute polarization calibration
8	**hifa_spwphaseup**	Compute wide to narrow spectral window mapping and per spectral window phase offsets
9	**hifa_timegaincal**	Determine temporal gains from calibrator observations
10	**hifa_tsyscal**	Derive a Tsys calibration table
11	hifa_tsysflagchans	Flag deviant channels in system temperature measurements
12	hifa_tsysflagspectra	Flag deviant system temperature measurements
13	**hifa_tsysflag**	Flag deviant system temperature measurements
14	**hifa_wvrgcalflag**	Calculate WVR corrections

15	**hifa_wvrgcal**	Compute the WVR calibration

5 Common Task Descriptions

5.1 h_init

h_init must be called before any other interferometry pipeline task. The pipeline can be initialised in one of two ways: by creating a new pipeline state (h_init) or be loading a saved pipeline state (h_resume). h_init creates an empty pipeline context but does not load visiblity data into the context. hif_importdata can be used to load interferometry data.

Task Description

Initialise the interferometry pipeline
The h_init task initialises the interferometry pipeline.

Keyword arguments:

---- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

---- pipeline context defined parameter argument which can be set only in 'interactive mode' or 'getinputs' modes

loglevel -- Pipeline log level threshold: (debug|info|warning|error|critical).
Log messages below this threshold will not be displayed.
default: 'info'

plotlevel -- Pipeline plot level threshold: (all|summary).
Switch between generation of all plots ('all') or just summary plots
('summary'), omitting the per antenna/spw/field plots and detail pages
from the web log.
default: 'all'

output_dir -- Working directory for pipeline processing. Some pipeline processing products such as HTML logs and images will be directed to subdirectories of this path.
default: './' (current directory)
overwrite -- Overwrite existing MSs on input.

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).

default: True

acceptresults -- Add the results
of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Examples

1. Create the pipeline context
h_init()

Parameter List

Table 5: h_init default settings

Parameter	Type	Default	Description
pipelinemode	string	automatic	The pipeline operating mode
loglevel	string	info	Log level for pipeline messages
plotlevel	string	all	Level for pipeline plots
output_dir	string	./	The output working directory
overwrite	bool	True	Overwrite existing files on import
dryrun bool	bool	False	Run the task (False) or display the task command (True)
acceptresults	bool	True	Add the results into the Pipeline context

5.2 h_resume

h_resume restores a named pipeline state from disk allowing a suspended pipeline reduction session to be resumed.

Task Description

Restore a save pipeline state from disk
h_resume restores a name pipeline state from disk allowing a suspended
pipeline reduction session to be resumed.

Keyword parameters:

filename -- Name of the saved pipeline state. Setting filename to 'last' restores the most recently saved pipeline state whose name begins with 'context*'.
default: 'last'
example: filename='context.s3.2012-02-13T10:49:11'
filename='last'

Examples

1. Resume the last saved session
h_resume()

2. Resume the named saved session
h_resume(filename='context.s3.2012-02-13T10:49:11')

Parameter List

Table 6: h_resume default settings

Parameter	Type	Default	Description
filename	string	last	Filename of saved state to be restored

5.3 h_save

h_save saves the current pipeline state to disk under a unique name. If no name is supplied one is generated automatically from a combination of the rootname 'context', the current stage number, and a timestamp.

Task Description

Save the pipeline state to disk
h_save saves the current pipeline state to disk under a unique name.

Keyword arguments:

filename -- Name of the saved pipeline state. If filename is '' then
a unique name will be generated computed from the root 'context', the
current stage number, and the timestamp.
default: ''

Examples

1. Save the current state in the default file
h_save()

2. Save the current state to a user named file
h_save(filename='savestate_1')

Parameter List

Table 7: h_save default settings

Parameter	Type	Default	Description
filename	string	None	Name for saved state

5.4 h_weblog

Task Description

Open the pipeline weblog in a browser

Parameter List

No parameters

6 Interferometry Common Task Descriptions

6.1 hif_antpos

The hif_antpos task corrects the antenna positions recorded in the ASDMs using updated antenna position calibration information determined after the observation was taken. Corrections can be input by hand, read from a file on disk, or in future by querying an ALMA database service. The corrections are used to generate a calibration table which is recorded in the pipeline context and applied to the raw visibility data, on the fly to generate other calibration tables, or permanently to generate calibrated visibilities for imaging.

Task Description

Derive an antenna position calibration table

Keyword arguments:

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context dependent pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.

---- pipeline parameter arguments which can be set in any pipeline mode

hm_antpos -- Heuristics method for retrieving the antenna position
corrections. The options are 'online' (not yet implemented), 'manual',
and 'file'.
default: 'manual'
example: hm_antpos='file'

antenna -- The list of antennas for which the positions are to be corrected
if hm_antpos is 'manual'
default: none
example 'DV05,DV07'

offsets -- The list of antenna offsets for each antenna in 'antennas'. Each
offset is a set of 3 floating point numbers separated by commas, specified
in the ITRF frame.
default: none
example: [0.01, 0.02, 0.03, 0.03, 0.02, 0.01]

antposfile -- The file(s) containing the antenna offsets. Used if hm_antpos is 'file'.

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- List of input visibility files
default: []
example: ['ngc5921.ms']
caltable -- Name of output gain calibration tables
default: []
example: caltable=['ngc5921.gcal']

-- Pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Examples

1. Correct the position of antenna 5 for all the visibility files in a single
pipeline run.
hif_antpos (antenna='DV05', offsets=[0.01, 0.02, 0.03])

2. Correct the position of antennas for all the visibility files in a single pipeline run using antenna positions files on disk. These files are assumed to conform to a default naming scheme if 'antposfile' is unspecified by the user.
hif_antpos (hm_antpos='file')

Parameter List

Table 8: hif_antpos default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets
caltable	stringArray	None	List of output caltable(s)
hm_antpos	string	manual	The antenna position determination method
antenna	string	None	List of antennas to be corrected
offsets	doubleArray	None	List of position corrections one set per antenna
antposfile	string	None	File containing antenna position corrections
pipelinemode	string	automatic	The pipeline operation mode
dryrun	bool	False	Run the task (False) or list commands(True)
acceptresults	bool	True	Automatically accept results into context

6.2 hif_applycal

hif_applycal applies the precomputed calibration tables stored in the pipeline context to the set of visibility files using predetermined field and spectral window maps and default values for the interpolation schemes. Users can interact with the pipeline calibration state using the tasks hif_export_calstate and hif_import_calstate.

Task Description

Apply the calibration(s) to the data
Apply precomputed calibrations to the data.

---- pipeline parameter arguments which can be set in any pipeline mode

applymode -- Calibration apply mode
''='calflagstrict': calibrate data and apply flags from solutions using
the strict flagging convention
'trial': report on flags from solutions, dataset entirely unchanged
'flagonly': apply flags from solutions only, data not calibrated
'calonly': calibrate data only, flags from solutions NOT applied
'calflagstrict':
'flagonlystrict':same as above except flag spws for which calibration is
unavailable in one or more tables (instead of allowing them to pass
uncalibrated and unflagged)
default: ''

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.

In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurement
sets in the pipeline context.
default: []
example: ['X227.ms']

field -- A string containing the list of field names or field ids to which
the calibration will be applied. Defaults to all fields in the pipeline
context.
default: ''
example: '3C279', '3C279, M82'

intent -- A string containing a the list of intents against which the
selected fields will be matched. Defaults to all supported intents
in the pipeline context.
default: ''
example: '*TARGET*'

spw -- The list of spectral windows and channels to which the calibration will be applied. Defaults to all science windows in the pipeline context.
default: ''
example: '17', '11, 15'

antenna -- The list of antennas to which the calibration will be applied. Defaults to all antennas. Not currently supported.

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False

acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned

Examples

1. Apply the calibration to the target data
hif_applycal (intent='TARGET')

Parameter List

Table 9: hif_applycal default settings

Parameter	Type	Default	Description
vis	stringArray	none	List of input measurement sets
field	string	none	Set of data selection field names or ids
intent	string	none	Set of data selection observing intents
spw	string	none	Set of data selection spectral window/channels
antenna	string	none	Set of data selection antenna ids
applymode	string	none	Calibration mode: ""="calflagstrict","calflag","calflagstrict","trial","flagonly","flagonlystrict", or "calonly"
calwt	boolArray	true	Calibrate the weights as well as the data
flagbackup	bool	true	Backup the flags before the apply
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run task (False) or display the command(True)
acceptresults	bool	True	Automatically accept results into the context

6.3 hif_atmflag

Spectral window channels with low atmospheric transmission are identified and flagged. The flagging view comprises a transmission spectrum for each spectral window calculated using the CASA atmosphere model. Flags are generated by running the following rules on each spectrum:
If flag_minabs = True then channels with transmission below fmin_limit are flagged.
If flag_nmedian = True then channels with transmission below fnm_limit * median transmission are flagged.
The flagging limits are set by frequency rather than by channel number.
The frequency frame is the native one of the spectral windows, usually TOPO.

Task Description

Flag channels with bad atmospheric transmission hif_atmflag flags channels where the atmospheric transmission is low

Keyword arguments:

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.

---- pipeline parameter arguments which can be set in any pipeline mode

intents -- Specifies the data intents whose channels are to be flagged
if they have low atmospheric transmission. This string is
inserted into the flagcmd given to the flagdata task
applying the flags; it must have a valid flagcmd format.
default '*AMP*,*BANDPASS*,*PHASE*'

flag_minabs -- True to flag channels with transmission < fmin_limit.
default False

fmin_limit -- The atmospheric transmission below which channels are to be
flagged if flag_minabs is True.
default 0.1
flag_nmedian -- True to flag channels with transmission < fnm_limit * median
transmission.
default: False

fnm_limit -- Flag channels with transmission < fnm_limit * median transmission,
if flag_nmedian is True.
default: 0.5

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- List of input measurement sets
default: [] - Use the measurement sets currently stored in the pipeline context.
example: vis=['X132.ms']

-- Pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True

acceptresults -- This parameter has no effect. The Tsyscal file is already
in the pipeline context and is flagged in situ.

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Examples

1. Flag channels with transmission below 0.1 in each SpW.
hif_atmflag()
equivalent to:
hif_atmflag(flag_minabs=True, fmin_limit=0.1)

2. Flag channels with transmission below 0.4 * median transmission across the spectral window.
hif_atmflag(flag_nmedian=True, fnm_limit=0.4)

Parameter List

Table 10: hif_atmflag default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets
intent	string	*AMP*,*BANDPASS*,*PHASE*	Data intents to which flags are to be applied
flag_minabs	bool	False	True to flag channels where transmission < fmin_limit
fmin_limit	double	0.1	Transmission limit below which channels are to be flagged
flag_nmedian	bool	False	True to flag channels where transmission < fnm_limit * median transmission
fnm_limit	double	0.5	If flag_nmedian then flag channels where transmission < fnm_limit *median transmission
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands(True)
acceptresults	bool	True	Automatically apply results to context

6.4 hif_bandpass

hif_bandpass computes a bandpass solution for every specified science spectral window. By default a 'phaseup' pre-calibration is performed and applied on the fly to to the data, before the bandpass is computed. The hif_refant task may be used to precompute a prioritized list of reference antennas.

Task Description

Compute bandpass calibration solutions
Compute amplitude and phase as a function of frequency for each spectral window in each measurement set.
Previous calibration can be applied on the fly.

Keyword arguments:

--- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.
phaseup -- Do a phaseup on the data before computing the bandpass solution
default: True

phaseupsolint -- The phase correction solution interval in CASA syntax.
Used when phaseup is True.
default: 'int'
example: 300

phaseupbw -- Bandwidth to be used for phaseup. Used when phaseup is True.
default: ''
example: '' default to entire bandpass, '500MHz' use central 500MHz

hm_bandtype -- The type of bandpass. The options are 'channel' and 'polynomial' for CASA bandpass types = 'B' and 'BPOLY' respectively.
solint -- Time and channel solution intervals in CASA syntax.
default: 'inf,7.8125MHz'
example: 'inf,10ch', 'inf'

maxchannels -- The bandpass solution smoothing factor in channels. The solution interval is bandwidth / 240. Set to 0 for no smoothing.
default: 240
example: 0
combine -- Data axes to combine for solving. Axes are '', 'scan','spw','field' or any comma-separated combination.
default; 'scan'

example: combine='scan,field'

minblperant -- Minimum number of baselines required per antenna for each solve
Antennas with fewer baselines are excluded from solutions. Used for
hm_bandtype='channel' only.
default: 4
minsnr -- Solutions below this SNR are rejected. Used for hm_bandtype=
'channel' only
default: 3.0

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurement
sets specified in the pipeline context.
default: ''
example: ['M51.ms']
caltable -- The list of output calibration tables. Defaults to the standard
pipeline naming convention.
default: ''
example: ['M51.bcal']

field -- The list of field names or field ids for which bandpasses are
computed. Defaults to all fields.
default: ''
example: '3C279', '3C279, M82'
intent -- A string containing a comma delimited list of intents against
which the the selected fields are matched. Defaults to all data
with bandpass intent.
default: ''
example: '*PHASE*'

spw -- The list of spectral windows and channels for which bandpasses are
computed. Defaults to all science spectral windows.
default: ''
example: '11,13,15,17'

refant -- Reference antenna names. Defaults to the value(s) stored in the
pipeline context. If undefined in the pipeline context defaults to
the CASA reference antenna naming scheme.
default: ''
example: refant='DV01', refant='DV06,DV07'
solnorm -- Normalise the bandpass solutions
default: False

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False

acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Examples

1. Compute a channel bandpass for all visibility files in the pipeline context using the CASA reference antenna determination scheme.
hif_bandpass()

2. Same as the above but precompute a prioritized reference antenna list
hif_refant()
hif_bandpass()

Parameter List

Table 11: hif_bandpass default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurment sets
caltable	stringArray	None	List of output caltables
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection intents
spw	string	None	Set of data selection spectral window/channels
antenna	string	None	Set of data selection antenna IDs
phaseup	bool	True	Phaseup before computing the bandpass
phaseupsolint	any	int	Phaseup correction solution interval
phaseupbw	string	None	Bandwidth to use for phaseup
hm_bandtype	string	channel	Bandpass solution type
solint	any	Inf	Solution intervals
maxchannels	int	240	The smoothing factor in channels
combine	string	scan	Data axes which to combine for solve (scan,

			spw, and/or field)
refant	string	None	Reference antenna names
solnorm	bool	True	Normalise the bandpass solution
minblperant	int	4	Minimum baselines per antenna required for solve
minsnr	double	3.0	Reject solutions below this SNR
degamp	variant	None	Degree for polynomial amplitude solution
degphase	variant	None	Degree for polynomial phase solution
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

6.5 hif_bpflagchans

Deviant channels in bandpass calibrations are detected by analysis of the bandpass calibration amplitudes.
Flags are generated by running the following rules on each spectrum:
If flag_diffmad is True then channels bracketing unusually large jumps
in the spectrum are detected and flagged.
If flag_tmf is True then all channels will be flagged if a substantial
proportion of channels have already been flagged for other reasons.
If flag_edges is True then channels at the band edges are detected and flagged.
If flag_sharps is True then channels covering sharp sepctral features are flagged and the flagging is extended in an attempt to cover the lower flanks of the features as well.

Task Description

Flag deviant channels in bandpass calibration
hif_bpflagchans flags deviant channels in the bandpass calibration table.

Keyword arguments:

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.

---- pipeline parameter arguments which can be set in any pipeline mode

flag_diffmad -- True to flag channels whose difference is greater than
diffmad_limit * the MAD of the channel to channel
differences across the spectrum.
default: True

diffmad_limit -- If flag_diffmad is True then channel pairs will be
flagged whose difference is greater than
diffmad_limit * the MAD of the difference spectrum.
default: 14

diffmad_nchan_limit -- If flag_diffmad is True and the number of channels
flagged by diffmad_limit exceeds diffmad_nchan_limit
then flag all channels.
default: 10000

flag_tmf -- True to flag all channels if the proportion of channels
already flagged is greater than tmf_frac_limit or if the
number of channels already flagged is greater than tmf_nchan_limit.
default: False

tmf_frac_limit -- If flag_tmf is True then all channels will be flagged if
proportionally more than tmf_frac_limit of them are
already flagged.
default: 0.05

tmf_nchan_limit -- If flag_tmf is True then all channels will be flagged if
more than tmf_nchan_limit of them are already flagged.
default: 0.05

flag_edges -- True to flag edges of spectra.
default: False
edge_limit -- The first channels inward from each end of the spectrum where the
channel to channel difference falls below 'edge_limit' times the
median across the spectrum are designated as the 'edges'. These
channels and those outside them are flagged. If the 'edges' lie more
than 1/4 of the way across the spectrum then no flagging is done.
default: 3

flag_sharps -- True to flag channels that cover sharp spectral features. Uses
the same algorithm as flag_sharps to flag the cores of the
features but extends the flagging over the feature flanks until
the channel to channel difference falls below 2 * the median
over the spectrum.
default: False

sharps_limit -- Flag as sharp feature cores those channels bracketing a channel
to channel difference > sharps_limit.

default: 0.05
---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

caltable -- List of input bandpass calibration tables
default: [] - Use the table currently stored in the pipeline context.

-- Pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True

acceptresults -- This parameter has no effect. The Tsyscal file is already in the pipeline context and is flagged in situ.

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Examples

1. Flag birdies in the bandpass calibration for each antenna/SpW.
Flag the entire calibration spectrum if more than 0.05 of channels or more than 4 channels in total are flagged.
hif_bpflagchans()
equivalent to:
hif_bpflagchans(flag_diffmad=True, diffmad_limit=7, diffmad_nchan_limit=4)

2. Flag birdies in the bandpass calibration for each antenna/SpW.
hif_bpflagchans(flag_tmf=False)
equivalent to:
hif_bpflagchans(flag_diffmad=True, diffmad_limit=7, diffmad_nchan_limit=10000)

Parameter List

Table 12: hif_bpflagchans default settings

Parameter	Type	Default	Description
caltable	stringArray	None	List of input caltables
flag_hilo	bool	True	True to flag outlier channels
fhl_limit	double	7	Flag channels further from median than limit * MAD

fhl_minsample	double	5	Minimum number of points in sample
flag_tmf	bool	True	True to flag all channels if proportion of channels flagged > tmf_limit
tmf_limit	double	0.3	Fraction of channels flagged that triggers flagging of all channels
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands(True)
acceptresults	bool	True	Automatically apply results to context

6.6 hif_cleanlist

Task Description
Compute clean map
Compute a cleaned image for a particular target source/intent and spectral window.

Keyword arguments:

--- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurment sets specified in the h_init or hif_importdata sets.
example: vis='ngc5921.ms'
vis=['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
default: use all measurement sets in the context

weighting -- Weighting to apply to visibilities:
default='natural'; example: weighting='uniform';
Options: 'natural','uniform','briggs',
'superuniform','briggsabs','radial'
weighting_robust -- For weighting='briggs' and 'briggsabs'
default=0.0; example: robust=0.5;
Options: -2.0 to 2.0; -2 (uniform)/+2 (natural)

weighting_noise -- For weighting='briggsabs' noise parameter to use for Briggs "abs" weighting example noise='1.0mJy'

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: False

acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Parameter List

Table 13: hif_cleanlist default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets
target_list	any	{}	Dictionary specifying targets to be imaged; blank will read list from context
weighting	string	natural	Weighting of uv (natural, uniform, briggs, ...)
robust	double	0.0	Briggs robustness parameter
noise	any	1.0Jy	noise parameter for briggs abs mode weighting
npixels	int	1	number of pixels for superuniform or briggs weighting
hm_masking	string	None	Pipeline heuristics masking option
hm_cleaning	string	None	Pipeline cleaning mode
tlimit	double	2.0 T	imes the sensitivity limit for cleaning
masklimit	int	4	Times good mask pixels for cleaning
maxncleans	int	1	Maximum number of

			clean task calls
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

6.7 hif_clean

Task Description

Compute clean map
Compute a cleaned image for a particular target source/intent and spectral window.

Keyword arguments:

--- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurement sets in the context.
default: ''
example: vis=['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']

imagename -- Prefix of output images. Defaults to one of the following options depending on the availability of project information.
'{ousstatus uid}.{field}.[{intent}.]s{stage number}.spw{spw}'
'multivis.{field}.[{intent}.]s{stage number}.spw{spw}'
cleanboxes and thresholds to use as it goes. For each iteration the output images are:
{prename}.iter{n}.image; cleaned and restored image
{prename}.iter{n}.psf; point spread function (dirty beam)
{prename}.iter{n}.flux; relative sky sensitivity over field
{prename}.iter{n}.flux.pbcoverage; relative pb coverage over field (only for mosaics)

{prename}.iter{n}.model; image of clean components
{prename}.iter{n}.residual; image of residuals
{prename}.iter{n}.cleanmask; image of cleanmask used
default: ''
example: 'test1'

intent -- An intent against which the selected fields are matched. Default
means select all data from fields specified by 'field' paramete
default: ''
example: '', 'TARGET'

field -- Fields id(s) or name(s) to image or mosaic. Must be set.
default:
example: '3C279', 'Centaurus*'

spw -- Spectral window/channels to image. \'\' for all science data.
default: ''
example: '9', '9,11'
mode -- Frequency imaging mode, 'mfs', 'frequency'. \'\' defaults to
'frequency' if intent parameter includes 'TARGET' otherwise 'mfs'.
default: ''
example: 'mfs', 'mosaic'
imagermode -- Advanced imaging mode e.g. mosaic or Cotton-Schwab clean.
Derived as follows:
1. The 'field' parameter is converted into a list of field_ids for each measurement set in 'vis'.
2. If there is more than 1 field_id in the list for any measurement set then imagermode is set to 'mosaic',
otherwise it will be set to 'csclean'.
default: ''

outframe -- The reference frame of the output image. The only supported option
is 'LSRK'
default: ''
example: 'LSRK'
imsize -- X and Y image size in pixels). Must be even and contain factors
2,3,5,7 only.
Default derived as follows:
1. Determine 'phasecenter' value and spread of field centres around it.
2. Set size of image to cover spread of field centres plus a border of
width 0.75 * beam radius (to first null).
3. Divide x and y extents by 'cell' values to arrive at the numbers of
pixels required.
default: ''
example: [320,320]

cell -- X and Y cell size. Derived from maximum UV spacing. Details TBD
default ''
example: ['0.5arcsec', '0.5arcsec']
phasecenter -- Direction measure or field id for the mosaic center.
Default derived as follows:

1. Make an array containing all the field centers to be imaged together.
2. Derive the mean direction from the directions array.
default: \'\'
example: 2

nchan -- Number of channels or planes in the output image, -1 for all
default: -1
example: 128
width -- Width of spectral dimension in frequency, \'\' for default.
default: \'\'
example: '7.8125MHz'

weighting -- Weighting to apply to visibilities. Options are: 'natural',
'uniform','briggs', 'superuniform','briggsabs','radial'
default='natural'
example: weighting='uniform'
robust -- Parameter for 'briggs' and 'briggsabs' weighting. Ranges from -2.0 to
2.0. -2 for uniform +2 for natural.
default=0.0
example: 0.5
noise -- Parameter for 'briggsabs' weighting
default: '1.0Jy'
example: '0.5Jy'
npixels -- Parameter for 'briggs' and 'briggsabs; weighting
default: 1
example: 1

restoringbeam -- Gaussian sestoring beam for clean, \'\' for default
default: \'\'
example:

hm_masking -- Clean masking mode. Options are 'none', 'centralquarter'
'psf', 'psfiter' and 'manual'
default: 'centralquarter'
example: 'manual'

mask -- Image mask for hm_masking manual mode. User responsible for
matching image sizes, coordinates, etc.
default: \'\'
example: 'mymask.mask'

niter -- Maximum number of iterations per clean call
default: 500
example: 100
threshold -- Threshold for cleaning
default: '0.0'
example: '0.05'

maxncleans -- Maximum number of clean calls

default: 1
example: 10

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Examples

Make an 'mfs' image of calibrator 3c279 using data in spectral window 1. The cell size is set to 0.2 arcsec in RA and Dec. Other clean parameters are derived from heuristics:
hif_clean(field='3c279', cell='0.2arcsec', spw='1', mode='mfs')
Make a cube of calibrator 3c279 using data in spectral window 1. The cube planes will be evenly spaced in frequency in the LSRK frame. Other clean parameters are derived from heuristics.
hif_clean(field='3c279', cell='0.2arcsec', spw='1', mode='frequency', outframe='LSRK')

Parameter List

Table 14: hif_clean default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets, \'\' for default
imagename	string	None	Prefix for image filenames, \'\' for default
intent	string	None	Set of data selection intents, \'\' for all
field	string	None	Set of data selection field names or ids
spw	string	None	Set of data selection spectral window/channels \'\' for all
uvrange	any	None	Set of uv ranges, \'\' for all

mode	string	None	Spectral gridding type (mfs, frequency, \'\' for default)
imagermode	string	None	Imaging mode (csclean, mosaic, \'\' for default)
outframe	string	None	velocity frame of output image (LSRK, \'\' for default)
imsize	intArray	None	Y and Y image size in pixels, single value same for both, \'\' for default
cell	stringArray	None	Y and Y cell size(s), single value same for both, \'\' for default
phasecenter	any	None	Image center (direction or field index), \'\' for default
nchan	int	-1	Number of channels or planes in output image, -1 = all
start	any	None	Start of output spectral dimension
width	any	None	Width of output spectral channels, \'\' for default
weighting	string	natural	Type of weighting
robust	double	0.0	Briggs weighting robustness parameter
noise	any	1.0Jy	Briggs weighting noise parameter
npixels	int	1	Weighting algorithm parameter
restoringbeam	stringArray	None	Gaussian restoring beam, \'\' for default
hm_masking	string	none	Pipeline heuristics masking option
hm_cleaning	string	manual	Pipeline clean control heuristics
mask	any	None U	ser mask, \'\' for whole image
niter	Int	500	Maximum number of clean iterations
threshold	double	0.0	Flux level to stop cleaning, must include units: \'1.0mJy\'
tlimit	double	2.0	Times the sensitivity limit for cleaning
masklimit	int	4	Times good mask pixels for cleaning

maxncleans	int	1	Maximum number of clean task calls
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

6.8 hif_export_calstate

hif_export_calstate saves the current pipeline calibration state to disk in the form of a set of equivalent applycal calls. If filename is not given, hif_export_calstate saves the calibration state to disk with a filename based on the pipeline context creation time, using the extension '.calstate'. One of two calibration states can be exported: either the active calibration state (those calibrations currently applied on-the-fly but scheduled for permanent application to the measurement set in a subsequent hif_applycal call) or the applied calibration state (calibrations that were previously applied to the measurement set using hif_applycal). The default is to export the active calibration state.

Task Description

Save the pipeline calibration state to disk
hif_export_calstate saves the current pipeline calibration state to disk
in the form of a set of equivalent applycal calls.

Keyword arguments:

filename -- Name for the saved calibration state.
state -- calibration state to export

Examples

1. Save the calibration state.
hif_export_calstate()

2. Save the active calibration state with a custom filename
hif_export_calstate(filename='afterbandpass.calstate')

3. Save the applied calibration state with a custom filename
hif_export_calstate(filename='applied.calstate', state='applied')

Parameter List

Table 15: hif_export_calstate default settings

Parameter	Type	Default	Description
filename	string	None	Name for saved calibration state

state	string	active	The calibration state to export

6.9 hif_exportdata

The hif_exportdata task exports the data defined in the pipeline context and exports it to the data products directory, converting and or packing it as necessary. The current version of the task exports the following products

- o an XML file containing the pipeline processing request
- o a tar file per ASDM / MS containing the final flags version
- o a text file per ASDM / MS containing the final calibration apply list
- o a FITS image for each selected calibrator source image
- o a FITS image for each selected science target source image
- o a tar file per session containing the caltables for that session
- o a tar file containing the file web log
- o a text file containing the final list of CASA commands
- o a python script that can be used to calibrate the data
- o a python script that can be used to restore raw data directly to calibrated measurement set

Task Description

Prepare interferometry data for export
The hif_exportdata task exports the data defined in the pipeline context and exports it to the data products directory, converting and or packing it as necessary.

Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In 'interactive' mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings
of all pipeline parameters without running the task.
default: 'automatic'.

---- pipeline context defined parameter argument which can be set only in 'interactive mode'

vis -- List of visibility data files for which flagging and calibration
information will be exported. Defaults to the list maintained in the
pipeline context.
default: []
example: vis=['X227.ms', 'X228.ms']

sessions -- List of sessions one per visibility file. Currently defaults
to a single virtual session containing al the visibility files in vis.
In future will default to set of observing sessions defined in the
context.
default: []

example: sessions=['session1', 'session2']

pprfile -- Name of the pipeline processing request to be exported. Defaults to a file matching the template 'PPR_*.xml'.
default: []
example: pprfile=['PPR_GRB021004.xml']

calintents -- List of calibrator image types to be exported. Defaults to all standard calibrator intents 'BANDPASS', 'PHASE', 'FLUX'
default: ''
example: calintents='PHASE'

calimages -- List of calibrator images to be exported. Defaults to all calibrator images recorded in the pipeline context.
default: []
example: calimages=['3C454.3.bandpass', '3C279.phase']

targetimages -- List of science target images to be exported. Defaults to all science target images recorded in the pipeline context.
default: []
example: targetimages=['NGC3256.band3', 'NGC3256.band6']

products_dir -- Name of the data products subdirectory. Defaults to './'
default: ''
example: products_dir='../products'

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True

acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Examples

1. Export the pipeline results for a single sessions to the data products directory
!mkdir ../products
hif_exportdata (products_dir='../products')

2. Export the pipeline results to the data products directory specify that

only the gain calibrator images be saved.
!mkdir ../products
hif_exportdata (products_dir='../products', calintents='*PHASE*')

Parameter List

Table 16: hif_exportdata default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input visibility data
sessions	stringArray	None	List of sessions one per visibility file
pprfile	string	None	The pipeline processing request file to be exported
calintents	string	None	The calibrator source target intents to be exported
calimages	stringArray	None	List of calibrator images to be exported
targetimages	stringArray	None	List of target images to be exported
products_dir	string	None	The data products directory
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline con

6.10 hif_gaincal

The complex gains are derived from the data column (raw data) divided by the model column (usually set with hif_setjy). The gains are obtained for a specified solution interval, spw combination and field combination. Good candidate reference antennas can be determined using the hif_refant task. Previous calibrations that have been stored in the pipeline context are applied on the fly. Users can interact with these calibrations via the hif_export_calstate and hif_import_calstate tasks.

Task Description

Determine temporal gains from calibrator observations
Compute the gain solutions.

---- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In interactive mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.

hm_gtype -- The type of gain calibration. The options are 'gtype' and
'gspline' for CASA gain types = 'G' and 'GSPLINE' respectively.
calmode -- Type of solution, The options are 'ap' (amp and phase), 'p'
(phase only) and 'a' (amp only)
default: 'ap'
options: 'p','a','ap'

solint -- Time solution intervals in CASA syntax. Works for hm_gtype='gtype' only.
default: 'inf'
example: 'inf', 'int', '100sec'
combine -- Data axes to combine for solving. Options are '','scan','spw',field'
or any comma-separated combination. Works for hm_gtype='gtype' only.
default; ''
example: combine=''

minblperant -- Minimum number of baselines required per antenna for each solve
Antennas with fewer baaselines are excluded from solutions. Works for
hm_gtype='gtype' only.
default: 4
example: minblperant=2
minsnr -- Solutions below this SNR are rejected. Works for hm_gtype=
'channel' only
default: 3.0

splinetime -- Spline timescale (sec). Used for hm_gtype='gspline'. Typical
splinetime should cover about 3 to 5 calibrator scans.
default: 3600 (1 hour)
example: splinetime=1000
npointaver -- Tune phase-unwrapping algorithm. Used for hm_gtype='gspline'
default: 3 (Keep at this value)
phasewrap -- Wrap the phase for changes larger than this amount (degrees)
Used for hm_gtype='gspline'.
default: 180 (Keep at this value)

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurment
sets specified in the pipeline context
default: ''
example: ['M82A.ms', 'M82B.ms']
caltable -- The list of output calibration tables. Defaults to the standard

pipeline naming convention.
default: ''
example: ['M82.gcal', 'M82B.gcal']
field -- The list of field names or field ids for which gain solutions are
to be computed. Defaults to all fields with the standard intent.
default: ''
example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against
which the the selected fields are matched. Defaults to *PHASE*.
default: ''
example: '', '*AMP*,*PHASE*'
spw -- The list of spectral windows and channels for which gain solutions are
computed. Defaults to all science spectral windows.
default: ''
example: '3C279', '3C279, M82'
smodel -- Point source Stokes parameters for source model (experimental)
Defaults to using standard MODEL_DATA column data.
default: []
example: [1,0,0,0] (I=1, unpolarized)

refant -- Reference antenna name(s) in priority order. Defaults to most recent
values set in the pipeline context. If no reference antenna is defined in
the pipeline context use the CASA defaults.
default: ''
example: refant='DV01', refant='DV05,DV07'
solnorm -- Normalise the gain solutions
default: False
--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned

Examples

1. Compute standard per scan gain solutions that will be used to calibrate
the target.
hif_gaincal()

Parameter List

Table 17: hif_gaincal default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets
caltable	stringArray	None	List of output caltables
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection observing intents
spw	string	None	Set of data selection spectral window/channels
antenna	string	None	Set of data selection antenna ids
hm_gaintype	string	gtype	The gain solution type (gtype or gpsline)
calmode	string	ap	Type of solution" (ap, p, a)
solint	any	inf	Solution intervals
combine	string	None	Data axes which to combine for solve (scan, spw, and/or field)
refant	string	None	Reference antenna names
solnorm	bool	False	Normalize average solution amplitudes to 1.0
minblperant	int	4	Minimum baselines per antenna required for solve
minsnr	double	3.0 R	eject solutions below this SNR
smodel	doubleArray	None	Point source Stokes parameters for source model
splinetime	double	3600.0	Spline timescale(sec)
npointaver	int	3	The phase-unwrapping algorithm
phasewrap	double	180.0	Wrap the phase for jumps greater than this value (degrees)
pipelinemode	string	automatic	The pipeline operating mode

dryrun	bool	False	Run task (False) or display the command(True)
acceptresults	bool	True	Automatically accept results into the context

6.11 hif_import_calstate

hif_import calstate clears and then recreates the pipeline calibration state based on the set of applycal calls given in the named file. The applycal statements are interpreted in additive fashion; for identically specified data selection targets, caltables specified in later statements will be added to the state created by earlier calls.

Task Description

Import a calibration state from disk
Import a calibration state to disk.

Keyword arguments:

filename -- Name of the saved calibration state.

Examples

1. Import a calibration state from disk.
hif_import_calstate(filename='aftergaincal.calstate')

Parameter List

Table 18: hif_import_calstate default settings

Parameter	Type	Default	Description
filename	string	None	Name of the saved calibration state

6.12 hif_importdata

Task Description

Imports data into the interferometry pipeline
The hif_importdata task loads the specified visibility data into the pipeline context unpacking and / or converting it as necessary.

Keyword arguments:

---- pipeline parameter arguments which can be set in any pipeline mode

vis -- List of visisbility data files. These may be ASDMs, tar files of ASDMs,

MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']

session -- List of sessions to which the visibility files belong. Defaults
to a single session containing all the visibility files, otherwise
a session must be assigned to each vis file.
default: []
example: session=['Session_1', 'Sessions_2']

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.

---- pipeline context defined parameter argument which can be set only in 'interactive mode'

asis -- ASDM tables to convert as is
default: 'Antenna Station Receiver CalAtmosphere'
example: 'Receiver', ''
process_caldevice -- Ingest the ASDM caldevice table
default: False
example: True
overwrite -- Overwrite existing MSs on output.
default: False
bdfflags -- Apply BDF flags on import
default: True

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.

Examples

1. Load an ASDM list in the ../rawdata subdirectory into the context.

hif_importdata (vis=['../rawdata/uid___A002_X30a93d_X43e',
'../rawdata/uid_A002_x30a93d_X44e'])

2. Load an MS in the current directory into the context.
hif_importdata (vis=[uid___A002_X30a93d_X43e.ms])

3. Load a tarred ASDM in ../rawdata into the context.
hif_importdata (vis=['../rawdata/uid___A002_X30a93d_X43e.tar.gz'])

4. Check the hif_importdata inputs, then import the data
myvislist = ['uid___A002_X30a93d_X43e.ms', 'uid_A002_x30a93d_X44e.ms']
hif_importdata(vis=myvislist, pipelinemode='getinputs')
hif_importdata(vis=myvislist)

5. Load an ASDM but check the results before accepting them into the context.
results = hif_importdata (vis=['uid___A002_X30a93d_X43e.ms'],
acceptresults=False)
results.accept()

6. Run in dryrun mode before running for real
results = hif_importdata (vis=['uid___A002_X30a93d_X43e.ms'], dryrun=True)
results = hif_importdata (vis=['uid___A002_X30a93d_X43e.ms'])

Parameter List

Table 19: hif_importdata default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input visibility data
session	stringArray	None	List of visibility data sessions
pipelinemode	string	automatic	The pipeline operating mode
asis	string	Antenna Station Receiver CalAtmosphere	ASDM to convert as is
process_caldevice	bool	False	Import the caldevice table from the ASDM
overwrite	bool	False	Overwrite existing files on import
bdfflags	bool	True	Apply BDF flags on import
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline Context

6.13 hif_lowgainflag

Task Description

Flag antennas with low or high gain hif_lowgainflag flags data for antennas with unusually low or high gains.

Keyword arguments:

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

---- pipeline parameter arguments which can be set in any pipeline mode

intent -- The data intent to be checked for antennas with low gain. The
default is blank, which causes the task to select the first
intent in the list ['BANDPASS', 'PHASE', 'AMPLITUDE'] for which
data are found.
default ''
flag_nmedian -- True to flag figures of merit greater than
fnm_hi_limit * median or lower than fnm_lo_limit * median.
default True
fnm_lo_limit -- Points lower than fnm_lo_limit * median are flagged.
default 0.7
fnm_hi_limit -- Points greater than fnm_hi_limit * median are flagged.
default 1.3

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

-- Pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True

acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
Deviant antennas are detected by analysis of a view showing their calibration
gains. This view is a list of 2D images with axes 'Time' and
'Antenna'; there is one image for each spectral window and intent.
A flagcmd to flag all data for an antenna will be generated by any gain that

is outside the range [fnm_lo_limit * median, fnm_hi_limit * median].

Parameter List

Table 20: hif_lowgainflag default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets
intent	string	None	Data intent whose gains are to checked
spw	string	None	Spectral window ids whose gains are to be checked
refant	string	None	Reference antenna names
flag_nmedian	bool	True	True to flag values outside range [fnm_lo_limit * median, fnm_hi_limit*nmedian]
fnm_lo_limit	double	0.7	Flag values lower than fnm_lo_limit * median
fnm_hi_limit	double	1.3	Flag values higher than fnm_hi_limit * median
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands(True)
acceptresults	bool	True	Automatically apply results to context

6.14 hif_makecleanlist

Generate a list of images to be cleaned. By default the list will include one image per science target per spw. Calibrator targets can be selected by setting appropriate values for itent. By default the output image cellsize is set to the minimum cell size consistent with the UV coverage. By default the image size in pixels is set to values determined by the cell size and the single dish beam size. If a calibrator is being imaged (intents 'PHASE', 'BANDPASS', 'FLUX' or 'AMPLITUDE') then the image dimensions are limited to 'calmaxpix' pixels. By default science target images are cubes and calibrator target images are single channel. Science target images may be mosaics or single fields.

Task Description

Compute list of clean images to be produced
Create a list of images to be cleaned.

Keyword Arguments

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In 'interactive' mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

--- pipeline parameter arguments which can be set in any pipeline mode

mode -- Frequency specification:
default: 'mfs'
example:
mode='mfs' produce one image from all specified data
mode='frequency', channels are specified in frequency
mode='velocity', channels are specified in velocity
cell -- Cell size (x, y)
default '' Compute cell size based on the UV coverage of all the fields to be imaged.
example: ['0.5arcsec', '0.5arcsec']

imsize -- Image X and Y size in pixels. The sizes must be even and divisible by 2,3,5,7 only.
default: '' The default values are derived as follows:
1. Determine phase center and spread of field centers around it.
2. Set the size of the image to cover the spread of field centers plus a border of width 0.75 * beam radius, to first null.
3. Divide X and Y extents by cell size to arrive at the number of pixels required.
example: [120, 120]

calmaxpix -- Maximum image X or Y size in pixels if a calibrator is being imaged ('PHASE', 'BANDPASS', 'AMPLITUDE' or 'FLUX').
default: 300
example: 300
width -- Output channel width.
default: '' Difference in frequency between first 2 selected channels. for frequency mode images.
example: '24.2kHz'

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the h_init or hif_importdata sets.
default '': use all measurement sets in the context
example: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
intent -- Select intents for which associated fields will be imaged.
default: 'TARGET'
example: 'PHASE,BANDPASS'

field -- Select fields to image. Use field name(s) NOT id(s). Mosaics
are assumed to have common source / field names. If intent is specified
only fields with data matching the intent will be selected. The fields
will be selected from measurement sets in 'vis'.
default: '' Fields matching matching intent, one image per target source.
example: '3C279', 'Centaurus*', '3C279,J1427-421'

spw -- Select spectral window/channels to image.
default: '' Individual images will be computed for all science spectral
windows.
example: '9'

uvrange -- Select a set of uv ranges ro image.
default: '' All uv data is included
example: '0~1000klambda', ['0~100klambda', 100~1000klambda]

phasecenter -- Direction measure or field id of the image center.
default: '' The default phase center is set to the mean of the field
directions of all fields that are to be image together.
example: 0, 'J2000 19h30m00 -40d00m00'

nchan -- Total number of channels in the output image(s)
default: -1 Selects enough channels to cover the data selected by
spw consistent with start and width.
example: 100
start -- First channel for frequency mode images.
default '' Starts at first input channel of the spw.
example: '22.3GHz'

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True

Output

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.

Examples

1. Make a list of science target images to be cleaned, one image per science
spw.
hif_makecleanlist()

2. Make a list of PHASE and BANDPASS calibrator targets to be imaged,
one image per science spw.
hif_makecleanlist(intent='PHASE,BANDPASS')

3. Make a list of PHASE calibrator images observed in spw 1, images limited to
50 pixels on a side.
hif_makecleanlist(intent='PHASE',spw='1',calmaxpix=50)

Parameter List

Table 21: hif_makecleanlist default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets
imagename	string	None	Prefix for output image names, \'\' for default.
intent	string	TARGET	Set of data selection intents
field	string	None	Set of data selection field names or ids, \'\' for all
spw	string	None	Set of data selection spectral window/channels, \'\' for all
uvrange	string	None	Set of data selection uv ranges, \'\' for all.
mode	string	None	Spectral gridding type (mfs, frequency, \'\' for default)
outframe	string	None	velocity frame of output image (LSRK, \'\' for default)
imsize	intArray	None	Image X and Y size(s) in pixels, \'\' for default. Single value same for both.
cell	stringArray	None	Image X and Y cell size(s), \'\' for default. Single value same for both
calmaxpix	int	300	Maximum X and Y size of calibrator images in pixels
phasecenter	any	None	Image center (direction or field index, \'\' for default)
nchan	int	-1	Number of channels, -1 =

			all
start	any	None	Channel start, \'\' for default
width	any	None	Channel width, \'\' for default.
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

6.15 hif_normflux

Derive flux densities for point source transfer calibrators using flux models for reference calibrators.
Flux values are determined by:

- o computing complex gain phase only solutions for all the science spectral windows using the calibrator data selected by the 'reference' and 'refintent' parameters and the 'transfer' and 'transintent' parameters, and the value of the 'phaseupsolint' parameter.
- o computing complex amplitude only solutions for all the science spectral windows using calibrator data selected with 'reference' and 'refintent' parameters and the 'transfer' and 'transintent' parameters, the value of the 'solint' parameter.
- o transfering the flux scale from the reference calibrators to the transfer calibrators using refspwmap for windows without data in the reference calibrators
- o extracted the computed flux values from the CASA logs and inserting them into the MODEL_DATA column.

Note that the flux corrected calibration table computed internally is not used in later pipeline apply calibration steps.

Task Description

Average calibrator fluxes across measurement sets
Derive flux densities for point source transfer calibrators using flux models
for reference calibrators.

---- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In
'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the users can check the settings
of all pipeline parameters without running the task.
default: 'automatic'.

phaseupsolint -- Time solution intervals in CASA syntax for the phase solution.
default: 'inf'
example: 'inf', 'int', '100sec'
solint -- Time solution intervals in CASA syntax for the amplitude solution.
default: 'inf'
example: 'inf', 'int', '100sec'

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurment sets specified in the pipeline context
default: ''
example: ['M32A.ms', 'M32B.ms']

reference -- A string containing a comma delimited list of field names defining the reference calibrators. Defaults to field names with intent '*AMP*'.
default: ''
example: 'M82,3C273'

transfer -- A string containing a comma delimited list of field names defining the transfer calibrators. Defaults to field names with intent '*PHASE*'.
default: ''
example: 'J1328+041,J1206+30'

refintent -- A string containing a comma delimited list of intents used to select the reference calibrators. Defaults to *AMP*.
default: ''
example: '', '*AMP*'

refspwmap -- Vector of spectral window ids enabling scaling across spectral windows. Defaults to no scaling
default: [-1]
example: [1,1,3,3] (4 spws, reference fields in 1 and 3, transfer fields in 0,1,2,3

transintent -- A string containing a comma delimited list of intents defining the transfer calibrators. Defaults to *PHASE*.
default: ''
example: '', '*PHASE*'

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or

reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned

Examples

1. Compute flux flux values for the phase calibrator using model data from the amplitude calibrator.
hif_gfluxscale ()

Parameter List

Parameter	Type	Default	Description
vis	string	None	List of input measurements sets
refintent	string	None	Observing intent of reference fields
transintent	string	None	Observing intent of transfer fields
reference	variant	None	Reference calibrator field name(s)
transfer	variant	None	Transfer calibrator field name(s)
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display commands (True)
acceptresults	bool	True	Automatically accept results into context

6.16 hif_rawflagchans

The hif_rawflagchans task flags deviant baseline/channels in the raw data. The flagging views used are derived from the raw data for the specified intent - default the first of BANDPASS, PHASE and AMPLITUDE to have data. Bad baseline/channels are flagged for all intents, not just the one that is the basis of the flagging views. For each spectral window the flagging view is a 2d image with axes 'channel' and 'baseline'. The pixel for each channel,baseline is the time average of the underlying unflagged raw data. The baseline axis is labelled by numbers of form id1.id2 where id1 and id2 are the IDs of the baseline antennas. Both id1 and id2 run over all antenna IDs in the observation. This means that each baseline is shown twice but has the benefit that 'bad' antennas are easily identified by eye.

Three flagging methods are available:
If parameter flag_hilo is set True then outliers from the median of each flagging view will be flagged. If parameter flag_bad_quadrant is set True then a simple 2 part test is used to check for bad antenna quadrants and/or bad baseline quadrants. Here a 'quadrant' is defined simply as one quarter of the channel

axis. The first part of the test is to note as 'suspect' those points further from the view median than fbq_hilo_limit * MAD. The second part is to flag entire antenna/quadrants if their fraction of suspect points exceeds fbq_antenna_frac_limit. Failing that, entire baseline/quadrants may be flagged if their fraction of suspect points exceeds fbq_baseline_frac_limit. Suspect points are not flagged unless as part of a bad antenna or baseline quadrant.

Task Description

Flag deviant baseline/channels in raw data

Keyword arguments:

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.

---- pipeline parameter arguments which can be set in any pipeline mode

flag_hilo -- True to flag channel/baseline data further from the view
median than fhl_limit * MAD.
default: True

fhl_limit -- If flag_hilo is True then flag channel/baseline data
further from the view median than fhl_limit * MAD.
default: 20

fhl_minsample -- Do no flagging if the view median and MAD are derived
from fewer than fhl_minsample view pixels.
default: 5

flag_bad_quadrant -- True to search for and flag bad antenna quadrants
and baseline quadrants. Here a /'quadrant/' is one quarter of the channel axis.
default: True

fbq_hilo_limit -- If flag_bad_quadrant is True then channel/baselines
further from the the view median than fbq_hilo_limit * MAD will be noted as 'suspect'.
If there are enough of them to indicate that an antenna or baseline
quadrant is bad then all channel/baselines in that quadrant will be flagged.
default: 8.0

fbq_antenna_frac_limit -- If flag_bad_quarant is True and the fraction
of suspect channel/baselines in a particular antenna/quadrant exceeds fbq_antenna_frac_limit
then all data for that antenna/quadrant will be flagged.
default: 0.2

fbq_baseline_frac_limit -- If flag_bad_quarant is True and the fraction

of suspect channel/baselines in a particular baseline/quadrant exceeds fbq_baseline_frac_limit then all data for that baseline/quadrant will be flagged.
default: 1.0 (i.e. no flagging)

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- List of input measurement sets.
default: [] - Use the measurement sets currently known to the pipeline context.

-- Pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True

acceptresults -- This parameter has no effect. The Tsyscal file is already in the pipeline context and is flagged in situ.

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Examples:

1. Flag bad quadrants and wild outliers, default method.
hif_rawflagchans()
equivalent to:
hif_rawflagchans(flag_hilo=True, fhl_limit=20,
flag_bad_quadrant=True, fbq_hilo_limit=8,
fbq_antenna_frac_limit=0.2, fbq_baseline_frac_limit=1.0)

Parameter List

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets
spw	string	None	Set of data selection spectral windows, \'\' for all
intent	string	None	Data selection intent: default is first of BANDPASS, PHASE or AMPLITUDE to have data
flag_hilo	bool	True	True to flag outlier baseline/channels

fhl_limit	double	20.0	Flag baseline/channels further from median than limit * MAD
fhl_minsample	double	5	Minimum number of points in sample
flag_bad_quadrant	bool	True	True to flag bad quadrants in antennas
fbq_hilo_limit	double	8.0	Note as \'suspect\' baseline/channels further from median than limit * MAD
fbq_antenna_frac_limit	double	0.2	Flag antenna quadrant if its fraction of \'suspect\' baseline/channels > limit
fbq_baseline_frac_limit	double	1.0	Flag baseline quadrant if its fraction of \'suspect\' baseline/channels > limit
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands(True)
acceptresults	bool	True	Automatically apply results to context

6.17 hif_refant

The hif_refant task selects a list of reference antennas and stores them in the pipeline context in priority order. The priority order is determined by a weighted combination of scores derived by the antenna selection heuristics. In manual mode the reference antennas can be set by hand.

Task Description

Select the best reference antennas
The hif_refant task selects a list of reference antennas and outputs them
in priority order. The priority order is determined by a weighted combination
of scores derived by the antenna selection heuristics.

Keyword arguments:

---- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

hm_refant -- The heuristics method or mode for selection the reference
antenna. The options are 'manual' and 'automatic. In manual

mode a user supplied referenence antenna refant is supplied.
In 'automatic' mode the antennas are selected automatically.
default: 'automatic'

refant -- The user supplied reference antenna for 'manual' mode. If
no antenna list is supplied an empty list is returned.
default: ''
example: 'DV05'
geometry -- Score antenna by proximity to the center of the array. This
option is quick as only the ANTENNA table must be read.
default: True

flagging -- Score antennas by percentage of unflagged data. This option
requires computing flagging statistics.
default: True

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurement
sets in the pipeline context.
default: ''
example: ['M31.ms']
field -- The list of field names or field ids for which flagging scores
are computed if hm_refant='automatic' and flagging = True
default: ''
example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against
which the the selected fields are matched. Defaults to all supported
intents.
default: ''
example: '*BANDPASS*'

spw -- The list of spectral windows and channels for which flagging scores
are computed if hm_refant='automatic' and flagging = True.
default: ''
example: '11,13,15,17'

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.

Examples

1. Compute the references antennas to be used for bandpass and gain calibration.
hif_refant()

Parameter List

Table 22: hif_refant default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets
field	string	None	Set of data selection field names or ids
spw	string	None	Set of data selection spectral windows / channels
intent	string	None	Set of data selection intents
hm_refant	string	automatic	The referance antenna heuristics mode
refant	string	None	List of reference antennas
geometry	bool	True	Score by proximity to center of the array
flagging	bool	True	Score by percentage of good data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command (True)
acceptresults	bool	True	Add the results into the pipeline context

6.18 hif_restoredata

The hif_restoredata restores flagged and calibrated data from archived ASDMs and pipeline flagging and calibration data products. Pending archive retrieval support hif_restore data assumes that the required products are available in the rawdata_dir in the format produced by the hif_exportdata task. hif_restoredata assumes that the following entities are available in the raw data directory

- o the ASDMs to be restored
- o for each ASDM in the input list
- o a compressed tar file of the final flagversions file, e.g.

uid___A002_X30a93d_X43e.ms.flagversions.tar.gz
o a text file containing the applycal instructions, e.g.
uid___A002_X30a93d_X43e.ms.calapply.txt
o a compressed tar file containing the caltables for the parent session,
e.g. uid___A001_X74_X29.session_3.caltables.tar.gz
hif_restore data performs the followinga operations
o imports the ASDM(s))
o removes the default MS.flagversions directory created by the filler
o restores the final MS.flagversions directory stored by the pipeline
o restores the final set of pipeine flags to the MS
o restores the final calibration state of the MS
o restores the final calibration tables for each MS
o applies the calibration tables to each MS

Task Description

Restore flagged and calibration interferometry data from a pipeline run The hif_restoredata task restores flagged and calibrated measurements sets from archived ASDMs and pipeline flagging and calibration date products.

Keyword arguments:

---- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In 'interactive' mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings
of all pipeline parameters without running the task.
default: 'automatic'.

---- pipeline context defined parameter argument which can be set only in 'interactive mode'

vis -- List of raw visibility data files to be restored. Assumed to be
in the directory specified by rawdata_dir.
default: None
example: vis=['uid___A002_X30a93d_X43e']
session -- List of sessions one per visibility file.
default: None
example: session=['session_3']

products_dir -- Name of the data products directory. Currently not
used.
default: '../products'
example: products_dir='myproductspath'
rawdata_dir -- Name of the rawdata subdirectory.
default: '../rawdata'
example: rawdata_dir='myrawdatapath'

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Examples

1. Restore the pipeline results for a single ASDM in a single session
hif_restoredata (vis=['uid___A002_X30a93d_X43e'], session=['session_1'])

Parameter List

Parameter	Type	Default	Description
vis	stringArray	None	List of input visibility data
session	stringArray	None	List of sessions one per visibility file
products_dir	string	../products	The archived pipeline data products directory
copytoraw	bool	True	Copy calibration and flagging tables to raw data directory
rawdata_dir	string	../rawdata	The rawdata directory
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

6.19 hif_setjy

Fills the model column with the model visibilities

Task Description

Fill the model column with calibrated visibilities
Fills the model column with the model visibilities.

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.

---- pipeline parameter arguments which can be set in any pipeline mode

fluxdensity -- Specified flux density [I,Q,U,V] in Jy. Uses [1,0,0,0]
flux density for unrecognized sources, and standard flux densities for
ones recognized by 'standard', including 3C286, 3C48, 3C147, and several
planets, moons, and asteroids.
default=-1
example: [3.06,0.0,0.0,0.0]

reffile -- Path to a file containing flux densities for calibrators unknown to
CASA. Values given in this file take precedence over the CASA-derived values
for all calibrators except solar system calibrators. By default the path is
set to the CSV file created by h_importdata, consisting of catalogue fluxes
extracted from the ASDM.
default: ''
example: '', 'working/flux.csv'

spix -- Spectral index for fluxdensity S = fluxdensity * (freq/reffreq)**spix
Only used if fluxdensity is being used. If fluxdensity is positive, and
spix is nonzero, then reffreq must be set too. It is applied in the same
way to all polarizations, and does not account for Faraday rotation or
depolarization.
Default: 0

reffreq -- The reference frequency for spix, given with units. Provided
to avoid division by zero. If the flux density is being scaled by spectral
index, then reffreq must be set to whatever reference frequency is correct
for the given fluxdensity and spix. It cannot be determined from vis. On
the other hand, if spix is 0, then any positive frequency can be used and
will be ignored.
Default: '1GHz'
Examples: '86.0GHz', '4.65e9Hz'

scalebychan -- This determines whether the fluxdensity set in the model is
calculated on a per channel basis. If False then only one fluxdensity
value is calculated per spw.
default: True
standard -- Flux density standard, used if fluxdensity[0] less than 0.0. The
options are: 'Baars','Perley 90','Perley-Taylor 95', 'Perley-Taylor 99',
'Perley-Butler 2010' and 'Butler-JPL-Horizons 2010'.
default: 'Butler-JPL-Horizons 2012' for solar system object

'Perley-Butler 2010' otherwise

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurment
sets defined in the pipeline context.
default: []
example:

field -- The list of field names or field ids for which the models are
to be set. Defaults to all fields with intent '*AMPLITUDE*'.
default: ''
example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against
which the the selected fields are matched. Defaults to all data
with amplitude intent.
default: ''
example: '*AMPLITUDE*'

spw -- The list of spectral windows and channels for which bandpasses are
computed. Defaults to all science spectral windows.
default: ''
example: '11,13,15,17'

model -- Model image for setting model visibilities. Not fully
supported.
default: ''
example: see details in help for CASA setjy task

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True

acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.

Examples

1. Set the model flux densities for all the amplitude calibrators.
hif_setjy()

Parameter List

Table 23: hif_setjy default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets
field	string	None	List of field names or ids
intent	string	None	Observing intent of flux calibrators
spw	string	None	List of spectral window ids
model	string	None	File location for field model
reffile	string	None	Path to file with fluxes for non-solar system calibrators
fluxdensity	any	-1	Specified flux density [I,Q,U,V]; -1 will lookup values
spix	double	0.0	Spectral index of fluxdensity
reffreq	string	1GHz	Reference frequency for spix
scalebychan	bool	True	Scale the flux density on a per channel basis or else on a per spw basis
standard	variant	None	Flux density standard
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the commands(True)
acceptresults	bool	True	Automatically accept results into the context

6.20 hif_show_calstate

hif_show_calstate displays the current on-the-fly calibration state of the pipeline as a set of equivalent applycal calls.

Task Description

Show the current pipeline calibration state

Keyword arguments:

None

Parameter List

No parameters

7 Interferometry ALMA Task Descriptions

7.1 hifa_bandpass

hifa_bandpass computes a bandpass solution for every specified science spectral window. By default a 'phaseup' pre-calibration is performed and applied on the fly to to the data, before the bandpass is computed. hifa_bandpass supercedes hif_bandpass for ALMA data.

Task Description

Compute bandpass calibration solutions
Compute amplitude and phase as a function of frequency for each spectral
window in each measurement set.
Previous calibration can be applied on the fly.

Keyword arguments:

--- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.

hm_phaseup -- The pre-bandpass solution phaseup gain heuristics. The options
are 'snr' (compute solution required to achieve the specified SNR), 'manual'
(use manual solution parameters), and '' (none).
default: 'snr'
example: hm_phaseup='manual'

phaseupsolint -- The phase correction solution interval in CASA syntax.
Used when hm_phaseup='manual' or as a default if the hm_phasup='snr'
heuristic computation fails.
default: 'int'
example: phaseupsolint='300s'

phaseupbw -- Bandwidth to be used for phaseup. Defaults to 500MHz.
Used when hm_phaseup='manual'.
default: ''
example: '' default to entire bandpass, '500MHz' use centreal 500MHz

phaseupsnr -- The required SNR for the phaseup solution. Used only if

hm_phaseup='snr'
default: 20.0
example: phaseupsnr=10.0

phaseupnsols -- The minimum number of phaseup gain solutions. Used only if hm_phaseup='snr'.
default: 2
example: phaseupnsols=4

hm_bandpass -- The bandpass solution heuristics. The options are 'snr' (compute the solution required to achieve the specified SNR), 'smoothed' (simple smoothing heuristics), and 'fixed' (use the user defined parameters for all spws).

solint -- Time and channel solution intervals in CASA syntax.
default: 'inf' Used for hm_bandpass='fixed', and as a default for the 'snr' and 'smoothed' options.
default: 'inf,7.8125MHz'
example: solint='inf,10ch', solint='inf'

maxchannels -- The bandpass solution smoothing factor in channels. The solution interval is bandwidth / 240. Set to 0 for no smoothing.
Used if hm_bandpass='smoothed".
default: 240
example: 0
bpsnr -- The required SNR for the bandpass solution. Used only if hm_bandpass='snr'
default: 50.0
example: bpsnr=20.0

bpnsols -- The minimum number of bandpass solutions. Used only if hm_bandpass='snr'.
default: 8

hm_bandtype -- The type of bandpass. The options are 'channel' and 'polynomial' for CASA bandpass types = 'B' and 'BPOLY' respectively.
combine -- Data axes to combine for solving. Axes are '', 'scan','spw','field' or any comma-separated combination.
default; 'scan'
example: combine='scan,field'

minblperant -- Minimum number of baselines required per antenna for each solve
Antennas with fewer baselines are excluded from solutions. Used for hm_bandtype='channel' only.
default: 4

minsnr -- Solutions below this SNR are rejected. Used for hm_bandtype= 'channel' only
default: 3.0

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurment sets specified in the pipeline context.
default: ''
example: ['M51.ms']

caltable -- The list of output calibration tables. Defaults to the standard pipeline naming convention.
default: ''
example: ['M51.bcal']

field -- The list of field names or field ids for which bandpasses are computed. Defaults to all fields.
default: ''
example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against which the the selected fields are matched. Defaults to all data with bandpass intent.
default: ''
example: '*PHASE*'

spw -- The list of spectral windows and channels for which bandpasses are computed. Defaults to all science spectral windows.
default: ''
example: '11,13,15,17'

refant -- Reference antenna names. Defaults to the value(s) stored in the pipeline context. If undefined in the pipeline context defaults to the CASA reference antenna naming scheme.
default: ''
example: refant='DV01', refant='DV06,DV07'

solnorm -- Normalise the bandpass solutions
default: False

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: False

acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.

Examples

1. Compute a channel bandpass for all visibility files in the pipeline
context using the CASA reference antenna determination scheme.
hif_bandpass()
2. Same as the above but precompute a prioritized reference antenna list
hif_refant()
hif_bandpass()

Parameter List

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets
caltable	stringArray	None	List of output caltables
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection intents
spw	string	None	Set of data selection spectral windows/channels
antenna	string	None	Set of data selection antenna IDs
hm_phaseup	string	snr	Phaseup before computing the bandpass
phaseupsolint	any	int	Phaseup correction solution interval
phaseupbw	string	None	Bandwidth to use for phaseup
phaseupsnr	double	20.0	SNR for phaseup solution
phaseupsols	int	2	Minimum number of phaseup gain solutions
hm_bandpass	string	snr	Bandpass solution heuristics
solint	any	inf	Solution intervals
maxchannels	int	240	The smoothing factor in channels
bpsnr	double	50.0	SNR for bandpass solution
bpnsols	int	8	Minimum number of bandpass solutions

hm_bandtype	string	channel	Bandpass solution type
combine	string	scan	Data axes to combine for solve (scan, spw, and/or field)
refant	string	None	Reference antenna names
solnorm	bool	True	Normalise the bandpass solution
minblperant	int	4	Minimum baselines per antenna required for solve
minsnr	double	3.0	Reject solutions below this SNR
degamp	variant	None	Degree for polynomial amplitude solution
degphase	variant	None	Degree for polynomial phase solution
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command (True)
acceptresults	bool	True	Add the results to the Pipeline context

7.2 hifa_bpsolint

Task Description

Compute optimal bandpass calibration solution intervals
Compute the bandpass phaseup time solution interval and bandpass frequency solution interval.

---- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In interactive mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.

phaseupnsr -- The required phaseup gain time interval solution signal to noise
default: 20.0
example: phaseupsnr = 10.0
minphaseupints -- The minimum number of time intervals in the phaseup gain
solution.
default: 2

example: minphaseupints=4

bpnsr -- The required bandpass frequency interval solution signal to noise
default: 50.0
example: phaseupsnr = 20.0
minbpnchan -- The minimum number of frequency intervals in the bandpass solution.
default: 8
example: minbpnchans=16

hm_nantennas -- The heuristics for determines the number of antennas to use in the signal to noise estimate. The options are 'all' and 'unflagged'.
The 'unflagged' options is not currently supported.
default: 'all'
example: hm_nantennas='unflagged'

maxfracflagged -- The maximum fraction of an antenna that can be flagged before its is excluded from the signal to noise estimate.
default: 0.90
example: maxfracflagged=0.80

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the pipeline context
default: ''
example: ['M82A.ms', 'M82B.ms']

field -- The list of field names of sources to be used for signal to noise estimation. Defaults to all fields with the standard intent.
default: ''
example: '3C279'

intent -- A string the intent against which the the selected fields are matched. Defaults to 'BANDPASS'.
default: 'BANDPASS'
example: intent='PHASE'

spw -- The list of spectral windows and channels for which gain solutions are computed. Defaults to all the science spectral windows for which there are both 'intent' and TARGET intents.
default: ''
example: '13,15'

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).

default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned

Description

The phaseup gain time and bandpass frequency intervals are determined as
follows.

- o For each data set the list of source(s) to use for bandpass solution signal to noise estimation is compiled based on the values of the field, intent, and spw parameters.
- o Source fluxes are determined for each spw and source combination.
- o Fluxes in Jy are derived from the pipeline context.
- o Pipeline context fluxes are derived from the online flux calibrator catalog, the ASDM, or the user via the flux.csv file.
- o If no fluxes are available the task terminates.
- o Atmospheric calibration and observations scans are determined for each spw and source combination.
- o If intent is set to 'PHASE' are there are no atmospheric scans associated with the 'PHASE' calibrator, 'TARGET' atmospheric scans will be used in stead.
- o If atmospheric scans cannot be associated with any of the spw and source combinations the task terminates.
- o Science spws are mapped to atmospheric spws for each science spw and source combinations.
- o If mappings cannot be determined for any of the spws the task terminates
- o The median Tsys value for each atmospheric spw and source combination is determined from the SYSCAL table. Medians are computed first by channel, then by antenna, in order to reduce sensitivity to deviant values.
- o The science spw parameters, exposure time(s), and integration time(s) are determined.
- o The phase up time interval, in time units and number of integrations required to meet the phaseupsnr are computed, along with the phaseup sensitivity in mJy and the signal to noise per integration. Nominal Tsys and sensitivity values per receiver band provide by the ALMA project are used for this estimate.
- o Warnings are issued if estimated phaseup gain time solution would contain fewer than minphasupints solutions
- o The frequency interval, in MHz and number of channels required to meet the bpsnr are computed, along with the per channel sensitivity in mJy and the per channle signal to noise. Nominal Tsys and sensitivity values per receiver band provide by the ALMA project are used for this estimate.
- o Warnings are issued if estimated bandpass olution would contain fewer than

minbpnchan solutions

Examples

1. Estimate the phaseup gain time interval and the bandpass frequency interval required to match the desired signal to noise for bandpass solutions.
hifa_bpsolint()

Parameter List

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets
field	string	None	Set of data selection field names
intent	string	BANDPASS	Set of data selection observing intents
spw	string	None	Set of data selection spectral window IDs
phaseupsnr	double	20.0	The required bandpass phaseup signal to noise
minphaseupints	int	2	The minimum number of phaseup intervals in the time solution
bpsnr	double	50.0	The required bandpass frequency signal to noise
minbpchan	int	8	The minimum number of channels in the frequency solution
hm_nantennas	bool	all	The antenna selection heuristic (unsupported)
maxfracflagged	string	0.90	The maximum fraction of data flagged per antenna (unsupported)
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run task (False) or display the command (True)
acceptresults	bool	True	Automatically accept the results into the context

7.3 hifa_flagdata

The hifa_flagdata data performs basic flagging operations on a list of measurements including:

- o applying online flags
- o apply a flagging template (manual flagging provided by user)
- o autocorrelation data flagging
- o shadowed antenna data flagging
- o scan based flagging by intent or scan number
- o edge channel flagging

Task Description

Do basic flagging of a list of measurement sets
The hifa_flagdata data performs basic flagging operations on a list of measurement sets.

Keyword arguments:

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.

---- pipeline parameter arguments which can be set in any pipeline mode

autocorr -- Flag autocorrelation data.
default: True

shadow -- Flag shadowed antennas.
default: True

scan -- Flag a list of specifed scans.
default: True

scannumber -- A string containing a comma delimited list of scans to be
flagged.
example: '3,5,6'
default: ''

intents -- A string containing a comma delimited list of intents against
which the scans to be flagged are matched.
example: '*BANDPASS*'
default: 'POINTING,FOCUS,ATMOSPHERE,SIDEBAND'

edgespw -- Flag the edge spectral window channels.
default: True

fracspw -- Fraction of the baseline correlator TDM edge channels to be flagged.

default: 0.0625

fracspwfps -- Fraction of the ACS correlator TDM edge channels to be flagged.
default: 0.48387
online -- Apply the online flags.
default: True

fileonline -- File containing the online flags. These are computed by the h_init or hif_importdata data tasks. If the online flags files are undefined a name of the form 'msname_flagonline.txt' is assumed.
default: ''
template -- Apply flagging templates
default: True

filetemplate -- The name of an text file that contains the flagging template for RFI, birdies, telluric lines, etc. If the template flags files is undefined a name of the form 'msname_flagtemplate.txt' is assumed.
default: ''

hm_tbuff -- The heuristic for computing the default time interval padding parameter. The options are 'halfint' and 'manual'. In 'halfint' mode tbuff is set to half the maximum of the median integration time of the science and calibrator target observations.
default: 'halfint'

tbuff -- The time in seconds used to pad flagging command time intervals if hm_tbuff='manual'.
default: 0.0

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurement
defined in the pipeline context.
example:
default: ''
flagbackup -- Back up any pre-existing flags.
default: False

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True

acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.

Examples

1. Do basic flagging on a measurement set
hifa_flagdata()

2. Do basic flagging on a measurement set flaggin additional scans selected by number as well.
hifa_flagdata(scannumber='13,18')

Parameter List

Table 24: hifa_flagdata default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets to flag
autocorr	bool	True	Flag autocorrelation data
shadow	bool	True	Flag shadowed antennas
scan	bool	True	Flag specified scans
scannumber	string	None	List of scans to be flagged
intents	string	POINTING,FOCUS,ATMOSPHERE,SIDEBAND	List of intents of scans to be flagged
edgespw	bool	True	Flag edge channels
fracspw	double	0.0625	Fraction of baseline correlator edge channels to be flagged
fracspwfps	double	0.048387	Fraction of ACA correlator edge channels to be flagged
online	bool	True	Apply the online flags
fileonline	string	None	File of online flags to be applied
template	bool	True	Apply a flagging template
filetemplate	stringArray	None	File that contains the flagging template
hm_tbuff	string	halfint	The time buffer computation heuristic

tbuff	any	0.0	List of time buffers (sec) to pad timerange in flag commands
pipelinemode	string	automatic	The pipeline operating mode
flagbackup	bool	False	Backup prexistings flags before applying new ones
dryrun	bool	False	Run the task (False) or display the command (True)
acceptresults	bool	True	Add the results into the pipeline context

7.4 hifa_fluxcalflag

Search the builtin solar system flux calibrater line catalog for overlaps with the science spectral windows. Generate a list of line overlap regions and flagging commands.

Task Description

Locate line regions in solar system flux calibrator spws
Fills the model column with the model visibilities.

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.

---- pipeline parameter arguments which can be set in any pipeline mode

threshold -- If the fraction of an spw occupied by line regions is greater
then threshold flag the entire spectral window.

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurment
sets defined in the pipeline context.
default: []
example:

field -- The list of field names or field ids for which the models are
to be set. Defaults to all fields with intent 'AMPLITUDE'.
default: ''
example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against which the the selected fields are matched. Defaults to all data with amplitude intent.
default: ''
example: 'AMPLITUDE'

spw -- The list of spectral windows and channels for which bandpasses are computed. Defaults to all science spectral windows.
default: ''
example: '11,13,15,17'

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True

acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Examples

1. Locate known lines in any solar system object flux calibrators.
hifa_fluxcalflag()

Parameter List

Table 25: hifa_fluxcalflag default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets
field	string	None	List of field names or ids
intent	string	None	Observing intent of flux calibrators
spw	string	None	List of spectral window ids
pipelinemode	string	automatic	The pipeline operating mode
threshold	double	0.75	Threshold for flagging the entire spw

appendlines	bool	False	Append user defined line regions to the line dictionary
linesfile	string	None	File containing user defined lines
applyflags	bool	True	Apply the computed flag commands
dryrun	bool	False	Run the task (False) or display the commands(True)
acceptresults	bool	True	Automatically accept results into the context

7.5 hifa_fluxdb

The task hifa_fluxdb connects to the ALMA flux calibrator database.

Task Description

Connect to flux calibrator database
Connect to the ALMA flux calibrator database

Keyword arguments:

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

vis -- List of input visibility files
default: none; example: vis='ngc5921.ms'

-- Pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True

acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise

the results object for the pipeline task is returned.

Example

1. Connect to the ALMA flux calibrator database
hifa_fluxdb()

Parameter List

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurment sets
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands (True)
acceptresults	bool	True	Automatically apply results to context

7.6 hifa_gfluxscale

Derive flux densities for point source transfer calibrators using flux models
for reference calibrators. Flux values are determined by:

- o computing complex gain phase only solutions for all the science spectral windows using the calibrator data selected by the 'reference' and 'refintent' parameters and the 'transfer' and 'transintent' parameters, and the value of the 'phaseupsolint' parameter.
- o computing complex amplitude only solutions for all the science spectral windows using calibrator data selected with 'reference' and 'refintent' parameters and the 'transfer' and 'transintent' parameters, the value of the 'solint' parameter.
- o transfering the flux scale from the reference calibrators to the transfer calibrators using refspwmap for windows without data in the reference calibrators
- o extracted the computed flux values from the CASA logs and inserting them into the MODEL_DATA column.

Resolved calibrators are handled via antenna selection either automatically, hm_resolvedcals='automatic' or manually, hm_resolvedcals='manual'. In the former case antennas closer to the reference antenna than the uv distance where visibilities fall to 'peak_fraction' of the peak are used. In manual mode the antennas specified in 'antenna' are used. Note that the flux corrected calibration table computed internally is not currently used in later pipeline apply calibration steps.

Task Description

Derive flux density scales from standard calibrators
Derive flux densities for point source transfer calibrators using flux models
for reference calibrators.

---- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the users can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

phaseupsolint -- Time solution intervals in CASA syntax for the phase solution.
default: 'int'
example: 'inf', 'int', '100sec'
solint -- Time solution intervals in CASA syntax for the amplitude solution.
default: 'inf'
example: 'inf', 'int', '100sec'

minsnr -- Minimum signal to noise ratio for gain calibration solutions.
default: 2.0
example: 1.5, 0.0

hm_resolvedcals - Heuristics method for handling resolved calibrators. The options are 'automatic' and 'manual'. In automatic mode antennas closer to the reference antenna than the uv distance where visibilities fall to 'peak_fraction' of the peak are used. In manual mode the antennas specified in 'antenna' are used.

antenna -- A comma delimited string specifying the antenna names or ids to be used for the fluxscale determination. Used in hm_resolvedcals='manual' mode.
default: ''.
example: 'DV16,DV07,DA12,DA08'

peak_fraction -- The limiting UV distance from the reference antenna forantennas to be included in the flux calibration. Defined as the point where the calibrator visibilities have fallen to 'peak_fraction' of the peak value.

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurment sets specified in the pipeline context
default: ''
example: ['M32A.ms', 'M32B.ms']

reference -- A string containing a comma delimited list of field names defining the reference calibrators. Defaults to field names with intent '*AMP*'.
default: ''
example: 'M82,3C273'

transfer -- A string containing a comma delimited list of field names

defining the transfer calibrators. Defaults to field names with
intent '*PHASE*'.
default: ''
example: 'J1328+041,J1206+30'

refintent -- A string containing a comma delimited list of intents
used to select the reference calibrators. Defaults to *AMP*.
default: ''
example: '', '*AMP*'

refspwmap -- Vector of spectral window ids enabling scaling across
spectral windows. Defaults to no scaling
default: [-1]
example: [1,1,3,3] (4 spws, reference fields in 1 and 3, transfer
fields in 0,1,2,3

transintent -- A string containing a comma delimited list of intents
defining the transfer calibrators. Defaults to *PHASE*.
default: ''
example: '', '*PHASE*'

refant -- A string specifying the reference antenna(s). By default
this is read from the context.
default: ''
example: 'DV05'

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False

acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned

Examples

1. Compute flux flux values for the phase calibrator using model data from
the amplitude calibrator.
hifa_gfluxscale ()

Parameter List

Table 26: hifa_gfluxscale default settings

Parameter	Type	Default	Description
vis	string	None	List of input measurements sets
reference	variant	None	Reference calibrator field name(s)
transfer	variant	None	Transfer calibrator field name(s)
refintent	string	None	Observing intent of reference fields
transintent	string	None	Observing intent of transfer fields
refspwmap	IntArray	-1	Map accross spectral window boundaries
phaseupsolint	any	int	Phaseup correction solution interval
solint	any	Inf	Amplitude correction solution interval
minsnr	double	2.0	Minimum SNR for gain solutions
refant	string	None	The name or ID of the reference antenna
hm_resolvedcals	string	automatic	The resolved calibrators heuristics method
antenna	string	None	Antennas to be used in fluxscale
peak_fraction	double	0.2	Fraction of peak visibility at uv-distance limit of antennas to be used
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display commands (True)
acceptresults	bool	True	Automatically accept results into context

7.7 hifa_importdata

Task Description

Imports data into the interferometry pipeline
The hifa_importdata task loads the specified visibility data into the pipeline context unpacking and / or converting it as necessary.

Keyword arguments:

---- pipeline parameter arguments which can be set in any pipeline mode

vis -- List of visisbility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']

session -- List of sessions to which the visibility files belong. Defaults
to a single session containing all the visibility files, otherwise
a session must be assigned to each vis file.
default: []
example: session=['Session_1', 'Sessions_2']

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.

---- pipeline context defined parameter argument which can be set only in 'interactive mode'
asis -- ASDM tables to convert as is
default: 'Antenna Station Receiver Source CalAtmosphere CalWVR'
example: 'Receiver', ''
process_caldevice -- Ingest the ASDM caldevice table
default: False
example: True

overwrite -- Overwrite existing MSs on output.
default: False

bdfflags -- Apply BDF flags on line
default: True

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True

acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Examples

1. Load an ASDM list in the ../rawdata subdirectory into the context.
hifa_importdata (vis=['../rawdata/uid___A002_X30a93d_X43e', '../rawdata/uid_A002_x30a93d_X44e'])

2. Load an MS in the current directory into the context.
hifa_importdata (vis=[uid___A002_X30a93d_X43e.ms])

3. Load a tarred ASDM in ../rawdata into the context.
hifa_importdata (vis=['../rawdata/uid___A002_X30a93d_X43e.tar.gz'])

4. Check the hif_importdata inputs, then import the data
myvislist = ['uid___A002_X30a93d_X43e.ms', 'uid_A002_x30a93d_X44e.ms']
hifa_importdata(vis=myvislist, pipelinemode='getinputs')
hifa_importdata(vis=myvislist)

5. Load an ASDM but check the results before accepting them into the context.
results = hifa_importdata (vis=['uid___A002_X30a93d_X43e.ms'], acceptresults=False)
results.accept()

6. Run in dryrun mode before running for real
results = hifa_importdata (vis=['uid___A002_X30a93d_X43e.ms'], dryrun=True)
results = hifa_importdata (vis=['uid___A002_X30a93d_X43e.ms'])

Parameter List

Table 27: hifa_importdata default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input visibility data
session	stringArray	None	List of visibility data sessions
pipelinemode	string	automatic	The pipeline operating mode
asis	string	Antenna Station Receiver Source CalAtmosphere CalWVR	ASDM to convert as is
process_caldevice	bool	False	Import the caldevice table from the ASDM
overwrite	bool	False	Overwrite existing files on import

bdfflags	bool	False[1]	Apply BDF flags on import
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

7.8 hifa_linpolcal

Task Description

Compute polarization calibration
Compute a polarization calibration.

Keyword arguments:

--- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurement
sets in the context. CURRENTLY THE LIST MUST CONTAIN 1 MEASUREMENT SET.
default: ''
example: vis=['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False

acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True

Output:

[1] Correct at time of Pipeline Release, but a change to default True is anticipated

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.

Parameter List

Table 28: hifa_linpolcal default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets, \'\' for default
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection intents
g0table	string	None	Name of table holding G0 gain - not accounting for source pol
delaytable	string	None	Name of table holding cross-hand delay
xyf0table	string	None	Name of table holding residual X-Y phase spectrum and source Q and U
g1table	string	None	Name of table holding G1 gain - accounting for source pol
df0table	string	None	Name of table holding instrument polarization gain
refant	string	None	Reference antenna names
spw	string	None	Set of data selection spectral window/channels
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

7.9 hifa_spwphaseup

The spectral window mapping from narrow to wide spectral windows
is computed as follows.

- construct a list of the bandwidths of all the science spectral windows

o determine the maximum bandwidth in this list maxbandwidth
o for each science spectral window with bandwidth less than maxbandwidth
o construct a list of spectral windows with bandwidths greater than
minfracmaxbw * maxbandwidth
o select the spectral window in this list whose band center most closely
matches the band center of the narr spectral window
o prefentially match within the same baseband is samebb is True
otherwise map the science spectral window to itself
o Phase offsets per spw are determined by computing a phase only gain
calibration on the selected data, normally the high signal to noise
bandpass calibration observations, using the solution interval 'inf'.
o Store the spectral window map and the phase offset calibration table
in the pipeline context for use by later tasks.

Task Description

Compute narrow to wide band spw map and per spw phase offsets
Compute the gain solutions.

---- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In interactive mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.

maxnarrowbw -- The maximum bandwidth defining narrow spectral windows. Values
must be in CASA compatible frequency units.
default: '300MHz'
example: combine=''

minfracmaxbw -- The minimum fraction of the maximum bandwidth in the set of
spws to use for matching.
default: 0.8
example: minfracmaxbw=0.75

samebb -- Match within the same baseband if possible ?
default: True
example: samebb=False

combine -- Data axes to combine for solving. Options are '','scan','spw',field'
or any comma-separated combination.
default: ''
example: combine=''

minblperant -- Minimum number of baselines required per antenna for each solve

Antennas with fewer baaselines are excluded from solutions.
default: 4
example: minblperant=2

minsnr -- Solutions below this SNR are rejected.
default: 3.0

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the pipeline context
default: ''
example: ['M82A.ms', 'M82B.ms']

caltable -- The list of output calibration tables. Defaults to the standard pipeline naming convention.
default: ''
example: ['M82.gcal', 'M82B.gcal']

field -- The list of field names or field ids for which gain solutions are to be computed. Defaults to all fields with the standard intent.
default: ''
example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against which the the selected fields are matched. Defaults to 'BANDPASS'
default: 'BANDPASS'
example: intent='', intent='*PHASE*'

spw -- The list of spectral windows and channels for which gain solutions are computed. Defaults to all the science spectral windows.
default: ''
example: '13,15'

refant -- Reference antenna name(s) in priority order. Defaults to most recent values set in the pipeline context. If no reference antenna is defined in the pipeline context the CASA defaults are used.
default: ''
example: refant='DV01', refant='DV05,DV07'

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned

Examples

1. Compute the simple narrow to wide spw map and the per spw phase offsets.
hif_spwphaseup()

Parameter List

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets
caltable	stringArray	None	List of output caltables
field	string	None	Set of data selection field names or IDs
intent	string	BANDPASS	Set of data selection observing intents
spw	string	None	Set of data selection spectral window/channels
maxnarrowbw	string	300 MHz	The maximum bandwidth defining narrow spectral windows
minfracmaxbw	double	0.8	The minimum fraction of the maximum bandpass for spw matching
samebb	bool	True	Match within the same baseband if possible
combine	string	None	Data axes which to combine for solve (scan, spw, and/or field)
refant	string	None	Reference antenna names
minblperant	int	4	Minimum baselines per antenna required for solve
minsnr	double	3.0	Reject solutions below this SNR
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	False Run task (False) or display the command(True)

acceptresults	bool	True	Automatically accept results into the context

7.10 hifa_timegaincal

The complex gains are derived from the data column (raw data) divided by the model column (usually set with hif_setjy). The gains are obtained for the specified solution intervals, spw combination and field combination. One gain solution is computed for the science targets and one for the calibrator targets. Good candidate reference antennas can be determined using the hif_refant task. Previous calibrations that have been stored in the pipeline context are applied on the fly. Users can interact with these calibrations via the hif_export_calstate and hif_import_calstate tasks.

Task Description

Determine temporal gains from calibrator observations
Compute the gain solutions.

---- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

calsolint -- Time solution interval in CASA syntax for calibrator source solutions.
default: 'int'
example: 'inf', 'int', '100sec'
targetsolint -- Time solution interval in CASA syntax for target source solutions.
default: 'inf'
example: 'inf', 'int', '100sec'

combine -- Data axes to combine for solving. Options are '','scan','spw',field'
or any comma-separated combination.
default; ''
example: combine=''

minblperant -- Minimum number of baselines required per antenna for each solve
Antennas with fewer baaselines are excluded from solutions.
default: 4
example: minblperant=2

calminsnr -- Solutions below this SNR are rejected for calibrator solutions.
default: 2.0
targetminsnr -- Solutions below this SNR are rejected for science target solutions.
default: 3.0

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- The list of input measurement sets. Defaults to the list of measurment
sets specified in the pipeline context
default: ''
example: ['M82A.ms', 'M82B.ms']

calamptable -- The list of output diagnostic calibration amplitude tables for the
calibration targets.
Defaults to the standard pipeline naming convention.
default: ''
example: ['M82.gacal', 'M82B.gacal']

calphasetable -- The list of output calibration phase tables for the
calibration targets.
Defaults to the standard pipeline naming convention.
default: ''
example: ['M82.gcal', 'M82B.gcal']

amptable -- The list of output calibration amplitude tables for the
calibration and science targets.
Defaults to the standard pipeline naming convention.
default: ''
example: ['M82.gcal', 'M82B.gcal']

targetphasetable -- The list of output phase calibration tables for the science targets.
Defaults to the standard pipeline naming convention.
default: ''
example: ['M82.gcal', 'M82B.gcal']

field -- The list of field names or field ids for which gain solutions are
to be computed. Defaults to all fields with the standard intent.
default: ''
example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against
which the the selected fields are matched. Defaults to the
equivalent of 'AMPLITUDE,PHASE,BANDPASS'.
default: ''
example: '', 'PHASE'

spw -- The list of spectral windows and channels for which gain solutions are
computed. Defaults to all science spectral windows.
default: ''
example: '3C279', '3C279, M82'

smodel -- Point source Stokes parameters for source model (experimental)
Defaults to using standard MODEL_DATA column data.
default: []
example: [1,0,0,0] (I=1, unpolarized)

refant -- Reference antenna name(s) in priority order. Defaults to most recent values set in the pipeline context. If no reference antenna is defined in the pipeline context use the CASA defaults.
default: ''
example: refant='DV01', refant='DV05,DV07'

solnorm -- Normalise the gain solutions
default: False

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: False

acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned

Examples

1. Compute standard per scan gain solutions that will be used to calibrate the target.
hifa_timegaincal()

Parameter List

Table 29: hifa_timegaincal default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets
calamptable	stringArray	None	List of diagnostic output amplitude caltables for calibrator targets
calphasetable	stringArray	None	List of output phase caltables for calibrator targets
targetphasetable	stringArray	None	List of output phase caltables for science targets
amptable	stringArray	None	List of output amp caltables for science targets

field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection observing intents
spw	string	None	Set of data selection spectral window/channels
antenna	string	None	Set of data selection antenna ids
calsolint	any	Int	Phase solution interval for calibrator sources
targetsolint	any	inf	Phase solution interval for science target sources
combine	string	None	Data axes which to combine for solve (scan, spw, and/or field)
refant	string	None	Reference antenna names
solnorm	bool	False	Normalize average solution amplitudes to 1.0
minblperant	int	4	Minimum baselines per antenna required for solve
calminsnr	double	2.0	Reject solutions below this SNR for calibrator solutions
targetminsnr	double	3.0	Reject solutions below this SNR for science solutions
smodel	doubleArray	None	Point source Stokes parameters for source model
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run task (False) or display the command(True)
acceptresults	bool	True	Automatically accept results into the context

7.11 hifa_tsyscal

Task Description

Derive a Tsys calibration table
Derive the Tsys calibration for list of measurement sets

Keyword arguments:

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

chantol -- The tolerance in channels for mapping atmospheric calibration windows (TDM) to science windows (FDM or TDM)
default: 1
example: 5

---- pipeline parameter arguments which can be set in any pipeline mode

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

vis -- List of input visibility files
default: none; example: vis='ngc5921.ms'

caltable -- Name of output gain calibration tables
default: none; example: caltable='ngc5921.gcal'

-- Pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True

acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Examples

1. Compute the tsys calibration tables for a list of measurement sets
hif_tsyscal()

Parameter List

Table 30: hifa_tsyscal default settings

Parameter	Type	Default	Description

vis	stringArray	None	List of input measurement sets
caltable	stringArray	None	List of output caltable(s)
chantol	int	1	Tsys spectral window map channel tolerance
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands(True)
acceptresults	bool	True	Automatically apply results to context

7.12 hifa_tsysflagchans

Deviant channels in Tsys measurements are detected by analysis of a compressed view of the Tsys results. This view comprises a list of Tsys spectra whose details are determined by the metric parameter. Flags are generated by running the following rules on each spectrum:
If flag_edges = True then channels at the band edges are detected and flagged.
If flag_sharps = True then channels covering sharp spectral features are flagged and then the flagging is extended in an attempt to cover the lower flanks of the features as well.

Task Description

Flag deviant channels in system temperature measurements
hif_tsysflagchans flags deviant channels in the system temperature calibration table.

Keyword arguments:

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

---- pipeline parameter arguments which can be set in any pipeline mode

metric -- Metric to use in calculating the flagging views of the Tsys data.
Possible values are:
'median' - A list of flagging views will be generated, one for each SpW/intentgroup. Each view is the median spectrum from all the member spectra.
'antenna_median' - A flagging view is generated for each SpW/intentgroup/antenna. Each view is the median spectrum from all the member spectra.
'antenna_diff' - A flagging view is generated for each SpW/intentgroup/antenna. Each view is the difference

between the median of spectra for the antenna/SpW
and the median for the SpW as a whole.
default: 'median'

intentgroups -- How data from various intents are to be combined in
separate flagging views. Internally the intentgroups are stored in a
list of strings but, because of interface limitations, this list
must be input as a single string.
default: "['ATMOSPHERE']"
For each spw the default creates a Tsys median spectrum
from all spectra with ATMOSPHERE intent.

flag_edges -- True to flag edges of spectra.
default True

edge_limit -- The first channels inward from each end of the spectrum where the
channel to channel difference falls below 'edge_limit' times the
median across the spectrum are designated as the 'edges'. These
channels and those outside them are flagged. If the 'edges' lie more
than 1/4 of the way across the spectrum then no flagging is done.
default 3

flag_sharps -- True to flag channels that cover sharp spectral features.
Flags sharp feature 'cores' where the channel to channel
difference exceeds a given limit, then extends the flagging
over the feature flanks until the channel to channel difference
falls below 2 * the median over the spectrum.
default: False

sharps_limit -- Flag as sharp feature cores those channels bracketing
a channel to channel difference > sharps_limit.
default: 0.05

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

caltable -- List of input Tsys calibration tables
default: [] - Use the table currently stored in the pipeline context.
example: caltable=['X132.ms.tsys.s2.tbl']

-- Pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- This parameter has no effect. The Tsyscal file is already
in the pipeline context and is flagged in situ.

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.

Examples

1. Flag the edges of the Tsys measurements in each SpW.
hif_tsysflagchans()
equivalent to:
hif_tsysflagchans(flag_edges=True, edge_limit=3)

2. Flag the edges of the Tsys measurements in each SpW, and sharp spectral
features in the band.
hif_tsysflagchans(flag_sharps=True)
equivalent to:
hif_tsysflagchans(flag_edges=True, edge_limit=3, flag_sharps=True, sharps_limit=0.05)

3. Flag antenna-based 'glitches':
hif_tsysflagchans(metric='antenna_diff', flag_edges=False, flag_sharps=True)
equivalent to:
hif_tsysflagchans(metric='antenna_diff', flag_edges=False, flag_sharps=True, sharps_limit=0.05)

Parameter List

Table 31:hifa_tsysflagchans default settings

Parameter	Type	Default	Description
caltable	stringArray	None	List of input caltables
metric	string	median	Method used to judge the quality of a Tsys measurement
intentgroups	string	"ATMOSPHERE"	List of groups of intents for which views are to be created
flag_edges	bool	True	True to flag edge channels of Tsys spectra
edge_limit	double	3	Candidate edge channels have channel to channel difference > edge_limit * median across spectrum
flag_sharps	bool	False	True to flag channels covering sharp spectral features plus additional channels on their flanks
sharps_limit	double	0.05	Flag channels bracketing a channel to channel difference > sharps_limit
pipelinemode	string	automatic	The pipeline operations mode

dryrun	bool	False	Run the task (False) or list commands(True)
acceptresults	bool	True	Automatically apply results to context

7.13 hifa_tsysflagspectra

Deviant Tsys measurements are detected by analysis of a compressed view of the Tsys results. This view is a list of 2D images with axes 'Time' and 'Antenna' and value determined by the metric parameter. There is one image for each spectral window and polarization and intent group in the set: ['ATMOSPHERE'] (this default grouping of intents can be overridden using the parameter 'intentgroups').
If metric = 'median' then each pixel in each image is the median of the Tsys spectrum obtained at that point.
If metric = 'shape' then each pixel in each image is a measure of the difference in the shape of the Tsys spectrum there from the shape of the median of all the Tsys spectra contributing to the image. The shape metric is computed as:
shape = 100 * mean(abs(normalized Tsys - median normalized Tsys))
and a normalized array is:
normalized = array / median (array)
If metric = 'derivative' then each pixel in each image is 100 * the MAD (median absolute deviation) of the channel by channel derivative of the Tsys spectrum.
If metric = 'fieldshape' then each pixel in each image is a measure of the difference in the shape of the Tsys spectrum there from the shape of the median of all Tsys spectra for that antenna with refintent.
The shape metric is computed as:
shape = 100 * mean(abs(normalized Tsys - reference normalized Tsys))
and a normalized array is:
normalized = array / median (array)
Flags are generated by running the following rules on each image:
If flag_nmedian = True then pixels greater than fnm_limit * median
of all the pixels in the image are flagged.
If flag_hi = True then pixels are flagged if they are greater
than median + fhi_limit * MAD where median and MAD are computed for all
the pixels in the image. No flags are set if there are fewer than
fhi_minsample pixels in the sample.
If flag_maxabs = True then pixels are flagged if they have an absolute
value greater than fmax_limit.
If flag_minabs = True then pixels are flagged if they have an absolute
value less than fmin_limit.
If flag_tmf1 = True then antennas with too many flagged points as defined
by tmf1_limit are flagged entirely.

Task Description

Flag deviant system temperature measurements
hif_tsyscal flags deviant system temperature measurements in the system
temperature calibration table.

Keyword arguments:

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

---- pipeline parameter arguments which can be set in any pipeline mode

flag_nmedian -- True to flag figures of merit greater than fnm_limit * median.
default True

fnm_limit -- Points greater than fnm_limit * median of samples are flagged.
default 2

flag_hi -- True to flag high figure of merit outliers.
default: False

fhi_limit -- Sample points beyond fhi_limit * median absolute deviation (MAD) of the figure of merit are flagged.
default: 5.0

fhi_minsample -- If there are too few points in the sample the statistics are assumed to be unreliable and no flagging is done.
default: 5

flag_maxabs -- True to flag figures of merit with absolute value above fmax_limit.
default: False

fmax_limit -- Figures of merit with absolute value above fmax_limit are flagged.
default: 5.0

flag_minabs -- True to flag figures of merit with absolute value below fmin_limit.
default: False

fmin_limit -- Figures of merit with absolute value below fmin_limit are flagged.
default: 5.0

flag_tmf1 -- True to flag antennas that have too many flagged points as a function of time.
default: False

tmf1_axis -- Name of axis being flagged (Cannot be changed at present).
default: Time

tmf1_limit -- Fraction of flagged antennas points that triggers flagging the entire antenna.
default: 0.5

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'

caltable -- List of input Tsys calibration tables
default: [] - Use the table currently stored in the pipeline context.
example: caltable=['X132.ms.tsys.s2.tbl']

metric -- Method to use as a measure of the Tsys quality. Possible values are:
'median' - Each pixel in each flagging image is the median value of the unflagged channels in the associated Tsys spectrum.
'shape' - Each pixel in each flagging image is a measure of the difference in shape of the unflagged part of the associated normalised Tsys spectrum from the median of the normalised Tsys spectra contributing to the image. The shape metric is computed as:
shape = 100 * mean(abs(normalized Tsys - median normalized Tsys))
where a normalized array is:
normalized = array / median (array)
'fieldshape' - Each pixel in each flagging image is a measure of the difference in shape of the unflagged part of the associated normalised Tsys spectrum from the median of the normalised Tsys spectra for that antenna for refintent. The shape metric is computed as:
shape = 100 * mean(abs(normalized Tsys - reference normalized Tsys))
where a normalized array is:
normalized = array / median (array)
'derivative' - Each pixel in each flagging image is 100 * the MAD
(median absolute deviation) of the channel by channel
derivative of the associated normalised Tsys spectrum.
default: 'median'

intentgroups -- How the data from various intents are to be combined in
the flagging views. Internally the intentgroups are stored in a
list of strings but, because of interface limitations, this list
must be input as a single string.
default: "['ATMOSPHERE']"
For each spw/pol the default creates a separate 2d
(ANTENNA v TIME) array of the Tsys metric for all data with
ATMOSPHERE intent.
example: "['AMPLITUDE + BANDPASS + PHASE + TARGET']"
For each spw/pol this would create a 2d array of the Tsys metric
using data for all the listed intents combined - useful if you
are looking for changes that depend on intent.

refintent -- When metric='fieldshape' this specifies the data intent whose
Tsys provide the 'reference' shape for comparison.
default: 'BANDPASS'
example: "['AMPLITUDE + BANDPASS + PHASE + TARGET']"
The median of data with all these intents will be used as the
'reference'.

-- Pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but

do not execute (False).
default: True

acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Examples

1. Flag Tsys measurements where the Tsys median is greater than 2 * the median of this metric over the flagging image.
This is currently the default.
hif_tsysflagspectra()

2. Flag Tsys measurements whose 'shape' metric lies more than 5 * MAD above the median of this metric over the flagging image.
hif_tsysflagspectra(metric='shape', flag_hi=True, fhi_limit=5, flag_nmedian=False)

3. Flag Tsys measurements whose 'derivative' metric is greater than 2.
hif_tsysflagspectra(metric='derivative', flag_maxabs=True, fmax_limit=2, flag_nmedian=False)

Parameter List

Table 32: hifa_tsysflagspectra default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets (Not used)
caltable	stringArray	None	List of input caltables
metric	string	median	Method used to judge the quality of a Tsys measurement
intentgroups	string	"ATMOSPHERE"	List of groups of intents whose Tsys are to be compared with the reference
refintent	string	BANDPASS	Data intent that provides the reference shape for \'fieldshape\' metric
flag_hi	bool	False	True to flag high figure of merit outliers
fhi_limit	double	5.0	Flag figure of merit values higher than fhi_limit *MAD
fhi_minsample	int	5	Minimum number of samples for

			valid MAD estimate
flag_maxabs	bool	False	True to flag pixels with absolute value higher than fmax_limit
fmax_limit	double	5.0	Flag pixels whose absolute value is higher than this limit
flag_minabs	bool	False	True to flag pixels with absolute value lower than fmin_limit
fmin_limit	double	5.0	Flag pixels whose absolute value is lower than this limit
flag_nmedian	bool	True	True to flag figure of merit values higher than fnm_limit * median
fnm_limit	double	2.0	Flag figure of merit values higher than fnm_limit *median
flag_tmf1	bool	False	True to flag data with too many flagged samples
tmf1_axis	string	Time	Name of flagging samples axis
tmf1_limit	double	0.5	Proportion of samples flagged that triggers too many flags rule
niter	int	1	Number of view calculation/flag iterations
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands(True)
acceptresults	bool	True	Automatically apply results to context

7.14 hifa_tsysflag

Task Description

Flag deviant system temperature measurements
hif_tsysflag tries to flag all deviant system temperature measurements in the system temperature calibration table. It does this by running a sequence of flagging tests, each designed to look for a different type of error. The tests are:
1. Flag Tsys spectra with high median values by running sister task hifa_tsysflagspectra with metric='median', flag_nmedian=True, fnm_limit=fnm_limit.
2. Flag Tsys spectra with high median derivatives by running hifa_tsysflagspectra with metric='derivative', flag_maxabs=True, fmax_limit=fd_max_limit. This is meant to spot spectra that are 'ringing'.
3. Flag the edge channels of the Tsys spectra in each SpW by running hifa_tsysflagchans with intentgroups=['ATMOSPHERE','BANDPASS','AMPLITUDE'], flag_edges=True, edge_limit=fe_edge_limit.
4. Flag Tsys spectra whose shape is different from that associated with

the BANDPASS intent or which are associated with an antenna that has
been too heavily flagged already, by running hif_tsysflagspectra with
metric='fieldshape', refintent=ff_refintent, flag_maxabs=True,
fmax_limit=ff_max_limit, flag_tmf1=True, tmf1_axis='Antenna1',
tmf1_limit=ff_tmf1_limit.
5. Flag 'birdies' by running hif_tsysflagchans with metric='antenna_diff',
flag_sharps=True, sharps_limit=fb_sharps_limit.

Keyword arguments:

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.

---- pipeline context defined parameter arguments which can be set only in 'interactive mode'
caltable -- List of input Tsys calibration tables
default: [] - Use the table currently stored in the pipeline context.
example: caltable=['X132.ms.tsys.s2.tbl']
flag_nmedian -- True to flag Tsys spectra with high median value.
default: True

fnm_limit -- Flag spectra with median value higher than fnm_limit * median
of this measure over all spectra.
default: 2.0

flag_derivative -- True to flag Tsys spectra with high median derivative.
default: True

fd_max_limit -- Flag spectra with median derivative higher than
fd_max_limit * median of this measure over all spectra.
default: 5.0

flag_edgechans -- True to flag edges of Tsys spectra.
default: True

fe_edge_limit -- Flag channels whose channel to channel difference >
fe_edge_limit * median across spectrum.
default: 3.0

flag_fieldshape -- True to flag Tsys spectra with a radically different
shape to those of the ff_refintent.
default: True

ff_refintent -- Data intent that provides the reference shape for
'flag_fieldshape'.
default: BANDPASS

ff_max_limit -- Flag Tsys spectra with 'fieldshape' metric values > ff_max_limit.
default: 5.0

ff_tmf1_limit -- Flag all Tsys spectra for an antenna if proportion flagged already > ff_tmf1_limit.
default: 0.666

flag_birdies -- True to flag channels covering sharp spectral features.
default: True

fb_sharps_limit -- Flag channels bracketing a channel to channel difference > fb_sharps_limit.
default: 0.05

-- Pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True

acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Examples

1. Flag Tsys measurements using currently recommended tests:
hif_tsysflag()
2. Flag Tsys measurements using all recommended tests apart from that using the 'fieldshape' metric.
hif_tsysflag(flag_fieldshape=False)

Parameter List

Table 33: hifa_tsysflag default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input measurement sets (Not used)
caltable	stringArray	None	List of input caltables

flag_nmedian	bool	True	True to flag Tsys spectra with high median value
fnm_limit	double	2.0	Flag spectra with median > fnm_limit * median over all spectra
flag_derivative	bool	True	True to flag Tsys spectra with high median derivative
fd_max_limit	double	5.0	Flag spectra with median derivative higher than fd_max_limit * median of this measure over all spectra
flag_edgechans	bool	True	True to flag edges of Tsys spectra
fe_edge_limit	double	3.0	Flag channels whose channel to channel difference > fe_edge_limit * median across spectrum
flag_fieldshape	bool	True	True to flag Tsys spectra with a radically different shape to those of the ff_refintent
ff_refintent	string	BANDPASS	Data intent providing the reference shape for \'flag_fieldshape\'
ff_max_limit	double	5.0	Flag Tsys spectra with \'fieldshape\' metric > ff_max_limit
ff_tmf1_limit	double	0.666	Flag all Tsys spectra for an antenna if proportion flagged already ff_tmf1_limit
flag_birdies	bool	True	True to flag channels covering sharp spectral features
fb_sharps_limit	double	0.05	Flag channels bracketing a channel to channel difference > fb_sharps_limit
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands(True)
acceptresults	bool	True	Automatically apply results to context

7.15 hifa_wvrgcalflag

Task Description

First, generate a gain table based on the Water Vapour Radiometer data in each vis file.
Second, apply the wvr calibration to the data specified by 'flag_intent', calculate flagging 'views' showing the ratio phase-rms with wvr / phase-rms without wvr for each scan. A ratio < 1 implies that the phase noise is improved, a score > 1 implies that it is made worse.
Third, search the flagging views for antennas with anomalous high values. If any are found then recalculate the wvr calibration with the 'wvrflag' parameter set to ignore their data and interpolate results from other antennas according to 'maxdistm' and 'minnumants'.
Fourth, if the overall QA score for the final wvr correction of a vis file is greater than the value in 'accept_threshold' then make available the wvr calibration file for merging into the context and use in the subsequent reduction.

vis -- List of input visibility files
default: none, in which case the vis files to be used will be read
from the context.
example: vis=['ngc5921.ms']

caltable -- List of output gain calibration tables
default: none, in which case the names of the caltables will be
generated automatically.
example: caltable='ngc5921.wvr'

hm_toffset -- If 'manual', set the 'toffset' parameter to the user-specified value.
If 'automatic', set the 'toffset' parameter according to the
date of the measurement set; toffset=-1 if before 2013-01-21T00:00:00
toffset=0 otherwise.
default: 'automatic'
toffset -- Time offset (sec) between interferometric and WVR data
default: 0

segsource -- If True calculate new atmospheric phase correction
coefficients for each source, subject to the constraints of
the 'tie' parameter. 'segsource' is forced to be True if
the 'tie' parameter is set to a non-empty value by the
user or by the automatic heuristic.
default: True

hm_tie -- If 'manual', set the 'tie' parameter to the user-specified value.
If 'automatic', set the 'tie' parameter to include with the
target all calibrators that are within 15 degrees of it:
if no calibrators are that close then 'tie' is left empty.
default: 'automatic'

tie -- Use the same atmospheric phase correction coefficients when
calculating the wvr correction for all sources in the 'tie'. If 'tie'

is not empty then 'segsource' is forced to be True. Ignored unless
hm_tie='manual'.
default: []
example: ['3C273,NGC253', 'IC433,3C279']

sourceflag -- Flag the WVR data for these source(s) as bad and do not produce
corrections for it. Requires segsource=True
default: []
example: ['3C273']

nsol -- Number of solutions for phase correction coefficients during this
observation, evenly distributed in time throughout the observation. It
is used only if segsource=False because if segsource=True then the
coefficients are recomputed whenever the telescope moves to a new source
(within the limits imposed by 'tie').
default: 1
disperse -- Apply correction for dispersion
default: False

wvrflag -- Flag the WVR data for these antenna(s) as bad and replace its data
with interpolated values
default: []
example: ['DV03','DA05','PM02']

hm_smooth -- If 'manual' set the 'smooth' parameter to the user-specified value.
If 'automatic', run the wvrgcal task with the range of 'smooth' parameters
required to match the integration time of the wvr data to that of the
interferometric data in each spectral window.

smooth -- Smooth WVR data on this timescale before calculating the correction.
Ignored unless hm_smooth='manual'.
default: '1s'
scale -- Scale the entire phase correction by this factor.
default: 1

maxdistm -- Maximum distance in meters of an antenna used for interpolation
from a flagged antenna.
default: 500
example: 550

minnumants -- Minimum number of nearby antennas (up to 3) used for
interpolation from a flagged antenna.
default: 2
example: 3

flag_intent -- The data intent(s) on whose wvr correction results the search
for bad wvr antennas is to be based.
A 'flagging view' will be calculated for each specified intent, in each
spectral window in each vis file.

Each 'flagging view' will consist of a 2-d image with dimensions ['ANTENNA', 'TIME'], showing the phase noise after the wvr correction has been applied.
If flag_intent is left blank, the default, the flagging views will be derived from data with the default bandpass calibration intent i.e. the first in the list BANDPASS, PHASE, AMPLITUDE for which the measurement set has data.
default ''

qa_intent -- The list of data intents on which the wvr correction is to be tried as a means of estimating its effectiveness.
A QA 'view' will be calculated for each specified intent, in each spectral window in each vis file.
Each QA 'view' will consist of a pair of 2-d images with dimensions ['ANTENNA', 'TIME'], one showing the data phase-noise before the wvr application, the second showing the phase noise after (both 'before' and 'after' images have a bandpass calibration applied as well).
An overall QA score is calculated for each vis file, by dividing the 'before' images by the 'after' and taking the median of the result. An overall score of 1 would correspond to no change in the phase noise, a score > 1 implies an improvement.
If the overall score for a vis file is less than the value in 'accept_threshold' then the wvr calibration file is not made available for merging into the context for use in the subsequent reduction.
default: 'BANDPASS,PHASE'

qa_bandpass_intent -- The data intent to use for the bandpass calibration in the qa calculation. The default is blank to allow the underlying bandpass task to select a sensible intent if the dataset lacks BANDPASS data.
default: ''
accept_threshold -- The phase-rms improvement ratio (rms without wvr / rms with wvr) above which the wrvg file will be accepted into the context for subsequent application.
default: 1.0

Examples

1. Compute the WVR calibration for all the measurement sets.
hifa_wvrgcalflag (hm_tie='automatic')

Parameter List

Table 34: hifa_wvrgcalflag default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input visibility files
caltable	stringArray	None	List of output gain calibration tables

hm_toffset	string	automatic	Toffset computation heuristic method
toffset	double	0	Time offset (sec) between IF and WVR data
segsource	bool	True	Compute new coefficient calculation for each source
sourceflag	stringArray	None	Flag the WVR data for these source(s)
hm_tie	string	automatic	Tie computation heuristics method
tie	stringArray	None	Sources for which to use the same atmospheric phase correction coefficients
nsol	int	1	Number of solutions for phase correction coefficients
disperse	bool	False	Apply correction for dispersion
wvrflag	stringArray	None	Flag the WVR data for these antenna(s) replace with interpolated values
hm_smooth	string	automatic	Smoothing computation heuristics method
smooth	string	1s	Smooth WVR data on the given timescale before calculating the correction
scale	double	1.	Scale the entire phase correction by this factor
maxdistm	double	500.	Maximum distance (m) of an antenna used for interpolation for a flagged antenna
minnumants	int	2	Minimum number of near antennas (up to 3) required for interpolation
mingoodfrac	double	0.8	Minimum fraction of good data per antenna antenna
flag_intent	string	None	Data intents to use in detecting and flagging bad wvr antennas
qa_intent	string	BANDPASS,PHASE	Data intents to use in estimating the

			effectiveness of the wvr correction
qa_bandpass_intent	string	None	Data intent to use for the bandpass calibration in the qa calculation
accept_threshold	double	1.0	Improvement ratio (phase-rms without wvr / phase-rms with wvr) above which wvrg calibration file will be accepted
flag_hi	bool	True	True to flag high figure of merit outliers
fhi_limit	double	10.0	Flag figure of merit values higher than limit *MAD
fhi_minsample	int	5	Minimum number of samples for valid MAD estimate
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

7.16 hifa_wvrgcal

Task Description

Generate a gain table based on the Water Vapour Radiometer data in each vis file. By applying the wvr calibration to the data specified by 'qa_intent' and 'qa_spw', calculate a QA score to indicate its effect on interferometric data; a score > 1 implies that the phase noise is improved, a score < 1 implies that it is made worse. If the score is less than 'accept_threshold' then the wvr gain table is not accepted into the context for subsequent use.

vis -- List of input visibility files
default: none, in which case the vis files to be used will be read
from the context.
example: vis=['ngc5921.ms']
caltable -- List of output gain calibration tables
default: none, in which case the names of the caltables will be
generated automatically.
example: caltable='ngc5921.wvr'

hm_toffset -- If 'manual', set the 'toffset' parameter to the user-specified value.

If 'automatic', set the 'toffset' parameter according to the
date of the measurement set; toffset=-1 if before 2013-01-21T00:00:00
toffset=0 otherwise.
default: 'automatic'

toffset -- Time offset (sec) between interferometric and WVR data
default: 0

segsource -- If True calculate new atmospheric phase correction
coefficients for each source, subject to the constraints of
the 'tie' parameter. 'segsource' is forced to be True if
the 'tie' parameter is set to a non-empty value by the
user or by the automatic heuristic.
default: True

hm_tie -- If 'manual', set the 'tie' parameter to the user-specified value.
If 'automatic', set the 'tie' parameter to include with the
target all calibrators that are within 15 degrees of it:
if no calibrators are that close then 'tie' is left empty.
default: 'automatic'

tie -- Use the same atmospheric phase correction coefficients when
calculating the wvr correction for all sources in the 'tie'. If 'tie'
is not empty then 'segsource' is forced to be True. Ignored unless
hm_tie='manual'.
default: []
example: ['3C273,NGC253', 'IC433,3C279']

sourceflag -- Flag the WVR data for these source(s) as bad and do not produce
corrections for it. Requires segsource=True.
default: []
example: ['3C273']

nsol -- Number of solutions for phase correction coefficients during this
observation, evenly distributed in time throughout the observation. It
is used only if segsource=False because if segsource=True then the
coefficients are recomputed whenever the telescope moves to a new source
(within the limits imposed by 'tie').
default: 1

disperse -- Apply correction for dispersion
default: False

wvrflag -- Flag the WVR data for the listed antennas as bad and replace
their data with values interpolated from the 3 nearest antennas with
unflagged data.
default: []
example: ['DV03','DA05','PM02']

hm_smooth -- If 'manual' set the 'smooth' parameter to the user-specified value.
If 'automatic', run the wvrgcal task with the range of 'smooth' parameters
required to match the integration time of the wvr data to that of the
interferometric data in each spectral window.
smooth -- Smooth WVR data on this timescale before calculating the correction.
Ignored unless hm_smooth='manual'.
default: '1s'

scale -- Scale the entire phase correction by this factor.
default: 1
maxdistm -- Maximum distance in meters of an antenna used for interpolation
from a flagged antenna.
default: 500
example: 550

minnumants -- Minimum number of nearby antennas (up to 3) used for
interpolation from a flagged antenna.
default: 2
example: 3

qa_intent -- The list of data intents on which the wvr correction is to be
tried as a means of estimating its effectiveness.
A QA 'view' will be calculated for each specified intent, in each spectral
window in each vis file.
Each QA 'view' will consist of a pair of 2-d images with dimensions
['ANTENNA', 'TIME'], one showing the data phase-noise before the
wvr application, the second showing the phase noise after (both 'before'
and 'after' images have a bandpass calibration applied as well).
An overall QA score is calculated for each vis file, by dividing the
'before' images by the 'after' and taking the median of the result. An
overall score of 1 would correspond to no change in the phase noise,
a score > 1 implies an improvement.
If the overall score for a vis file is less than the value in
'accept_threshold' then the wvr calibration file is not made available
for merging into the context for use in the subsequent reduction.
If you do not want any QA calculations then set qa_intent=''.
default: ''
example: 'PHASE'

qa_bandpass_intent -- The data intent to use for the bandpass calibration
in the qa calculation. The default is blank to allow the
underlying bandpass task to select a sensible intent if the dataset
lacks BANDPASS data.
default: ''

qa_spw -- The SpW(s) to use for the qa calculation, in the order
that they should be tried. Input as a comma-separated list.
The default is blank, in which case the task will try SpWs
in order of decreasing median sky opacity.

default: ''

accept_threshold -- The phase-rms improvement ratio
(rms without wvr / rms with wvr) above which the wrvg file will be
accepted into the context for subsequent application.
default: 1.0

Examples

1. Compute the WVR calibration for all the measurement sets.
hifa_wvrgcal (hm_tie='automatic')

Parameter List

Table 35: hifa_wvrgcal default settings

Parameter	Type	Default	Description
vis	stringArray	None	List of input visibility files
caltable	stringArray	None	List of output gain calibration tables
hm_toffset	string	automatic	Toffset computation heuristic method
toffset	double	0	Time offset (sec) between IF and WVR data
segsource	bool	True	Compute new coefficient calculation for each source
sourceflag	stringArray	None	Flag the WVR data for these source(s)
hm_tie	string	automatic	Tie computation heuristics method
tie	stringArray	None	Sources for which to use the same atmospheric phase correction coefficients
nsol	int	1	Number of solutions for phase correction coefficients
disperse	bool	False	Apply correction for dispersion
wvrflag	stringArray	None	Flag the WVR data for these antenna(s) replace with interpolated values
hm_smooth	string	automatic	Smoothing computation heuristics method
smooth	string	1s	Smooth WVR data on the given timescale before calculating the

			correction
scale	double	1.	Scale the entire phase correction by this factor
maxdistm	double	500.	Maximum distance (m) of an antenna used for interpolation for a flagged antenna
minnumants	int	2	Minimum number of near antennas (up to 3) required for interpolation
mingoodfrac	double	0.8	Minimum fraction of good data per antenna
qa_intent	string	None	Data intents to use in estimating the effectiveness of the wvr correction
qa_bandpass_intent	string	None	Data intent to use for the bandpass calibration in the qa calculation
qa_spw	string	None	Data SpW(s) to use in estimating the effectiveness of the wvr correction
accept_threshold	double	1.0	Improvement ratio (phase-rms without wvr / phase-rms with wvr) above which wvrg calibration file will be accepted
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

The Atacama Large Millimeter/submillimeter Array (ALMA), an international astronomy facility, is a partnership of Europe, North America and East Asia in cooperation with the Republic of Chile. ALMA is funded in Europe by the European Organization for Astronomical Research in the Southern Hemisphere (ESO), in North America by the U.S. National Science Foundation (NSF) in cooperation with the National Research Council of Canada (NRC) and the National Science Council of Taiwan (NSC) and in East Asia by the National Institutes of Natural Sciences (NINS) of Japan in cooperation with the Academia Sinica (AS) in Taiwan. ALMA construction and operations are led on behalf of Europe by ESO, on behalf of North America by the National Radio Astronomy Observatory (NRAO), which is managed by Associated Universities, Inc. (AUI) and on behalf of East Asia by the National Astronomical Observatory of Japan (NAOJ). The Joint ALMA Observatory (JAO) provides the unified leadership and management of the construction, commissioning and operation of ALMA.